

(19) 日本国特許庁 (J P)

(12) 公表特許公報 (A)

(11) 特許出願公表番号  
特表2002-528932  
(P2002-528932A)

(43) 公表日 平成14年9月3日(2002.9.3)

(51) Int.Cl. <sup>7</sup>	識別記号	F I	テームト* (参考)
H 0 4 Q 3/545		H 0 4 Q 3/545	5 K 0 2 4
H 0 4 L 12/24		H 0 4 L 12/24	5 K 0 2 6
H 0 4 M 3/00		H 0 4 M 3/00	E 5 K 0 3 0
3/42		3/42	A 5 K 0 5 1

審査請求 未請求 予備審査請求 有 (全 104 頁)

(21) 出願番号 特願2000-577571(P2000-577571)  
(86) (22) 出願日 平成11年10月20日(1999.10.20)  
(85) 翻訳文提出日 平成13年4月20日(2001.4.20)  
(86) 国際出願番号 P C T / U S 9 9 / 2 4 5 8 6  
(87) 国際公開番号 W O 0 0 / 2 3 8 9 8  
(87) 国際公開日 平成12年4月27日(2000.4.27)  
(31) 優先権主張番号 6 0 / 1 0 4 , 8 9 0  
(32) 優先日 平成10年10月20日(1998.10.20)  
(33) 優先権主張国 米国 (U S)

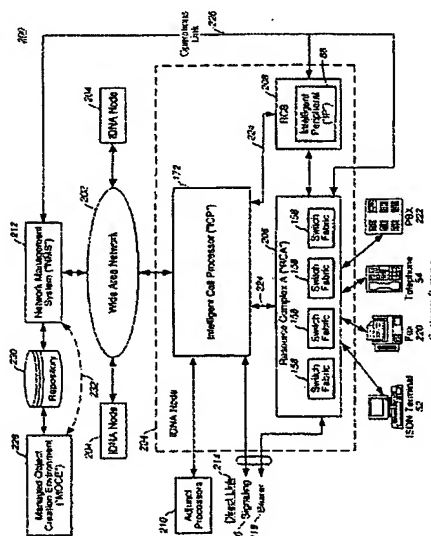
(71) 出願人 アジェイ・デオ  
アメリカ合衆国・テキサス・75056・ルイス  
スヴィル・サー・トリスタン・レーン・  
2508  
(71) 出願人 ウェンディ・ウォン  
アメリカ合衆国・テキサス・78287・ダラス・  
ケイブ・コーラル・ドライブ・4816  
(71) 出願人 ヘンリー・ワン  
アメリカ合衆国・コロラド・80919・コロラド・  
スプリングス・ガーデン・オブ・ザ・ガズ・  
ロード・2424  
(74) 代理人 弁理士 志賀 正武 (外7名)

最終頁に続く

(54) 【発明の名称】 インテリジェント・ネットワークにおけるリアルタイム呼処理サービスを提供する方法および装置

(57) 【要約】

インテリジェント・ネットワークにおいてリアルタイム呼処理サービスを提供するシステムであって、該システムは、インテリジェント・ネットワークにおけるサービス・ノード(204)において実行するオブジェクト・インスタンス間の通信を可能にするプラットフォームに依存しない通信システムを含む。発信側スイッチと関連した実行環境において実行するオペレーティング・システム・エージェント・オブジェクト(204)インスタンスは、スイッチで受け取った呼イベントに対応する発呼情報を、ネットワーク内のサービス・ノード(204)において提供される実行環境で実行する1つまたは複数のオブジェクト・インスタンスに送り、このオブジェクト・インスタンスは、発呼と関連した通信線路の状態を維持する線オブジェクト・インスタンスと、顧客要求にしたがってサービスを実行するメソッドを実施するサービス・オブジェクトとを含む。



(2)

**【特許請求の範囲】**

【請求項1】 それぞれ、メモリ記憶装置と、サービス・ノードに関連したネットワーク・スイッチ素子における事象の受信に応答して、サービスを実施するための実行環境を備えた、複数のサービス・ノードを有する通信ネットワークに関するサービス制御システムであって、

各ノード毎に、各ノードにおけるサービス処理に関連したサービス・オブジェクト資源のタイプ及び量を含むサービス・プロファイルを生成し、前記プロファイルに基づいて、サービス・オブジェクト資源の前記タイプ及び量を前記ノードにダウンロードするためのサービス・アドミニストレータと、

前記1つ以上の実行環境における実行に備えて、サービス・オブジェクトのインスタンス生成を行うためのインスタンス生成機構と、

サービス・ノードにおける実行環境資源を追跡して、前記ネットワークの各サービス・ノードにおいて利用可能なサービス・タイプのリストを保守し、各サービス・タイプの関連機能状況によって、要求されるサービスが、サービス・ノードにおけるインスタンス生成に利用可能であるか否かが表示されるようにするための資源管理装置とが含まれており、

前記機能状況によって、要求されるサービスが前記ネットワークにおけるインスタンス生成に利用不可能であることが表示されると、前記資源管理装置は、新しいサービス・オブジェクトのインスタンス生成が必要であることを前記中央アドミニストレータに伝え、サービス・ノードにおける新しいサービスのダウンロード及び起動を実施させることを特徴とするシステム。

【請求項2】 前記インスタンス生成機構に、

前記メモリ記憶システムから1つ以上のサービス・オブジェクトをロードし、前記実行環境内における実行に備えて、前記1つ以上のオブジェクトのインスタンス生成を行うための第1のオブジェクトと、

特定のサービスに対応し、そのサービスに関する各受信要求に対応する各サービス・インスタンスに1つ以上のサービス・スレッドを割り当て、各サービス・スレッド・インスタンスに、一意性の識別子が対応づけられるようにする第2の

(3)

オブジェクトと

が含まれることを特徴とする請求項1に記載のシステム。

【請求項3】 さらに、実行中のオブジェクト・インスタンス間においてメッセージ及び事象の実時間伝達を行うためのネットワーク・オペレーティング・システムが含まれることと、特定のサービスに対応する前記第2のオブジェクトが、前記サービス・インスタンス間における事象及びメッセージのチャネリングを行うことと、前記事象及びメッセージに、受信メッセージ及び事象を適正なサービス・インスタンスに対応づけるための前記一意性識別子が含まれることを特徴とする請求項2に記載のシステム。

【請求項4】 さらに、各サービス・スレッド・インスタンス毎に割り当てられた、サービス実行中に受信する前記サービス・インスタンスに対応づけられた事象を待ち行列に入れるための事象待ち行列機構が含まれ、

事象には、前記事象を実施すべき順番を指示する関連優先順位があることと、前記事象待ち行列装置が、その関連優先順位に基づいて受信事象の処理を可能にすることを特徴とする請求項3に記載のシステム。

【請求項5】 さらに、前記サービス・ノードに関する初期サービス機能を実施する構成ファイルに従って、まず、前記メモリ記憶システムから1つ以上のサービス・オブジェクトをロードするクラス・ローダ・プロセスが含まれることと、前記クラス・ローダが、事前定義サービス機能戦略に従って、利用可能な前記第1のオブジェクト及び任意のサービス・オブジェクトのインスタンス生成を行う責務を負うことを特徴とする請求項3に記載のシステム。

【請求項6】 特定のサービスに対応する前記第2のオブジェクトに、あるサービスに関連したスレッド・インスタンスの量と前記サービス・プロファイルにおいて求められる所定のしきい値を比較し、前記実行環境において、新しいサービス・スレッド・インスタンスのインスタンス生成がもはや支援されない場合、前記資源管理装置に対する警告信号を発生するためのスレッド・マネージャ・インスタンスが含まれることを特徴とする請求項3に記載のシステム。

【請求項7】 前記サービス・オブジェクト・インスタンス生成機構に、前記ネットワーク・オペレーティング・システムが含まれることと、前記資源管理

(4)

装置が、さらに、各サービス・ノードにおける実行環境の処理能力を追跡し、その処理能力に基づいて、あるサービス・ノードにおける実行環境で、あるサービスの実行が可能であるか否かを前記ネットワーク・オペレーティング・システムに表示することを特徴とする請求項6に記載のシステム。

【請求項8】 前記資源管理装置が、さらに、ある実行環境において現在実行中のサービス・スレッド数が、前記所定のしきい値を超えると、前記ネットワーク・オペレーティング・システムにオーバロード状況表示を伝達し、前記実行環境におけるそれ以上のサービス・オブジェクトのインスタンス生成を阻止することを特徴とする請求項7に記載のシステム。

【請求項9】 前記インスタンス生成機構に、  
各前記実行環境に設けられた実行環境で実行されるサービスのインスタンスに対応するアクティブ・サービス・オブジェクト・スレッドのレジストリと、  
オブジェクト参照を備えたサービス論理名をマッピングするためのマッピング装置と  
が含まれていることと、  
前記ネットワーク・オペレーティング・システムが、前記オブジェクト参照を利用して、局所実行環境において要求されるサービス・オブジェクト・スレッド・インスタンスのインスタンス生成を可能にすることを特徴とする請求項3に記載のシステム。

【請求項10】 各サービス・ノード毎に、メモリ記憶装置と、サービス・ノードに関連したネットワーク・スイッチ素子における事象の受信に応答して、サービスを実施するための実行環境が設けられた、通信ネットワークのサービス・ノードにおいてサービスを提供するための方法であって、

各サービス・ノード毎に、各ノードにおけるサービス処理に関連したサービス・オブジェクト資源のタイプ及び量を含むサービス・プロファイルを生成し、前記プロファイルに基づいて、サービス・オブジェクト資源の前記タイプ及び量を前記ノードにダウンロードするステップと、

前記1つ以上の実行環境における実行に備えて、サービス・オブジェクトのインスタンス生成を行うステップと、



(5)

各サービス・ノードにおいて利用可能なサービス・タイプのリストを保守することによって、サービス・ノードにおける実行環境資源を追跡し、各サービス・タイプの関連機能状況によって、要求されるサービスが、サービス・ノードにおけるインスタンス生成に利用可能であるか否かが表示されるようにするステップとが含まれており、

前記機能状況によって、要求されるサービスが前記ネットワークにおけるインスタンス生成に利用不可能であることが表示されると、新しいサービス・オブジェクトのインスタンス生成が必要であることを中央アドミニストレータ装置に伝え、サービス・ノードにおける新しいサービス・オブジェクトのダウンロード及び起動を実施させることを特徴とする方法。

【請求項11】 前記インスタンス生成ステップに、

受信したサービス要求に従って、前記メモリ記憶システムから1つ以上のサービス・オブジェクトをロードし、前記実行環境内における実行に備えて、前記1つ以上のオブジェクトのインスタンス生成を行うための第1のオブジェクトを設けるステップと、

特定のサービスに対応し、そのサービスに関する各受信要求に対応する各サービス・インスタンスに1つ以上のサービス・スレッドを割り当て、各サービス・スレッド・インスタンスに、一意性の識別子が対応づけられるようにする第2のオブジェクトを設けるステップと

が含まれることを特徴とする請求項10に記載の方法。

【請求項12】 さらに、サービス処理を支援して、1つ以上の実行中のサービス・オブジェクト間において、サービス・オブジェクトの実行中に生成されるメッセージ及び事象を伝達するステップが含まれることと、前記事象及びメッセージが、前記一意性識別子によって識別され、前記第2のオブジェクトを介して実行中のサービス・インスタンスが補正されることを特徴とする請求項11に記載の方法。

【請求項13】 さらに、サービス実行中に受信する、実行中のサービス・インスタンスに対応づけられた事象を待ち行列に入れるステップが含まれること

(6)

と、前記事象には、前記事象を実施すべき順番を指示する関連優先順位があることと、前記受信事象が、その対応する優先順位に基づいて処理されることを特徴とする請求項12に記載の方法。

【請求項14】 さらに、前記サービス・ノードに関する初期サービス機能を提供する構成ファイルに従って、まず、前記メモリ記憶システムから1つ以上のサービス・オブジェクトをロードするステップが含まれることと、前記クラス・ローダが、前記クラス・ローダが、事前定義サービス機能戦略に従って、利用可能な前記第1のオブジェクト及び任意のサービス・オブジェクトのインスタンス生成を行う責務を負うことを特徴とする請求項10に記載の方法。

【請求項15】 サービス・ノードにおける実行環境資源を追跡する前記ステップに、

あるサービスに関連したスレッド・インスタンスの量と前記サービス・プロファイルにおいて求められる所定のしきい値を比較するステップと、

前記実行環境において、新しいサービス・スレッド・インスタンスのインスタンス生成がもはや支援されない場合、資源管理装置に対する警告信号を発生するステップと

が含まれることを特徴とする請求項11に記載の方法。

【請求項16】 前記インスタンス生成機構に、

各前記実行環境に設けられた実行環境で実行されるサービスのインスタンスに対応するアクティブ・サービス・オブジェクト・スレッドのレジストリを保守するステップと、

オブジェクト参照を備えたサービス論理名をマッピングするステップと、

前記オブジェクト参照を利用して、局所実行環境において要求されるサービス・オブジェクト・スレッド・インスタンスのインスタンス生成を可能にするステップと

が含まれることを特徴とする請求項10に記載の方法。

(7)

## 【発明の詳細な説明】

## 【0001】

## 【発明の属する技術分野】

本発明は、一般に、通信サービスを提供するためのインテリジェント・ネットワーク・システムに関し、具体的には、インテリジェント・ネットワーク全体に分散した複数のサービス・ノードのそれぞれにおいてリアルタイムイベント処理サービスを提供する新規なサービス制御システムに関する。

## 【0002】

## 【従来の技術】

ネットワーク・サービスは、1つまたは複数の加入者との対話に応じて、データや電話などの通信ネットワークおよびそれと関連するリソースによって実行される機能である。たとえば、加入者が特別の順番の数字をダイヤルすることによって、自動転送や音声メール・アクセスなどの電話ネットワーク常駐サービスを起動することができる。セキュリティ、妥当性検査および認証により、その他のネットワーク・サービスをネットワーク所有者の支援に向けることができる。サービスの追加や修正は、通信ネットワークの変更を必要とする。

## 【0003】

最も慣例的な通信ネットワークは、相互接続されたスイッチと通信サービスから構成される。そのようなスイッチは、スイッチメーカーが設計した独自のソフトウェアまたはファームウェアによって動作する一体になった処理装置または埋め込まれた処理装置によって制御される。一般に、スイッチメーカーのソフトウェアまたはファームウェアは、サービス処理、呼処理、設備処理およびネットワーク管理の機能的なすべての側面をサポートしていなければならない。すなわち、ネットワーク所有者が新しいサービスを実現したり既存のサービスを修正したりしたいときは、様々なスイッチメーカーは、ネットワーク内のすべてのスイッチのソフトウェアを改訂しなければならない。

## 【0004】

ネットワークは様々なメーカーによる様々なスイッチモデルを含むため、新しいソフトウェアを慎重に開発し、試験し、展開することが必要になる。新しい改

(8)

訂ごとに各スイッチのコード・サイズが大きく複雑になってきているため、新しいソフトウェアを開発し試験し展開するのに必要な時間は、より長く複雑になってきている。したがって、このプロセスには、数年かかることがある。さらに、この高い複雑さによって、スイッチのプロセッサに負担がかかり、スイッチの動作不良の可能性が高まり、スイッチの修理と交換が必要になることがある。さらに、複数のネットワーク所有者が共通の組のスイッチメーカーに依存するため、競争を制限する望ましくない2つの状況が生じる。

#### 【0005】

第1に、メーカーのソフトウェア・リリースは、いくつかのネットワーク所有者から要求された変更を組み込もうとすることがあり、したがって、ネットワーク所有者は、自分たちのサービスとそれらの競争から提供されるサービスとを真に区別することができない。また、これにより、いくつかのネットワーク所有者は、メーカーが新しいリリースに他のネットワーク所有者による要求を組み込むまで待たされる。第2に、あるネットワーク所有者からの要求に応じた機能を組み込んで新しいサービスを実現したスイッチのソフトウェア・リリースは、意図せずに他のネットワーク所有者に利用可能になることがある。

#### 【0006】

このような問題は、加入者の移動性の高まり、トラヒックの種類と帯域幅の増大、伝統的な番号付け方式の崩壊、サービスの高度化、および競争の高まりにより、過去5～10年の間に新しいネットワーク・サービスの需要が指数的に高まってきたため許容できなくなってきた。したがって、新しいネットワーク・アーキテクチャが、サービス・ロジックを作成し展開し実行するよりフレキシブルな方法を取り入れる必要があることが広く認識されている。後で説明する本発明の新規なアーキテクチャを十分に理解するために、図1を参照して、関連する従来技術の以下の説明を提供する。

#### 【0007】

図1を参照し、本発明を含む様々なスイッチングアーキテクチャの論理表現を示す。全体を20で示した一体式スイッチは、サービス処理機能22、呼処理機能24、設備処理機能26およびスイッチ機構28を含む。これらの機能22、

(9)

24、26および28はすべて、グループ30で記号化したようにハードコードされ、混合され、画一化されている。さらに、機能22、24、26および28は、スイッチメーカーによって設計され、メーカーにより異なる独自のプラットフォーム上で動作する。その結果、これらの機能22、24、26および28は、メーカーのサポートなしに修正できず、これが、サービスの開発と実装を遅延させ、新しいサービスを市場に出すコストを高めている。したがって、新しい革新的なサービス、呼処理、データ処理、信号処理およびネットワーク操作の開発は、メーカー独自のスイッチハードウェアおよびソフトウェアに対するメーカーの管理と、業界標準を確立し実現する本来の難しさによる制限を受ける。

#### 【0008】

サービス処理機能22は、一体式スイッチ20内でコード化され、ローカルのデータ内容とダイヤルされた番号に基づいたプロセスのローカル制御しか可能にしない。このローカル情報は、コード化されたサービス機能を実行するハードコードされたプロセス・エンジンによって解釈される。呼処理機能24は、ハードコードされ、発呼機能と着呼機能を提供する。このプロセスは、実際に、個々の接続を確立して呼を完了させる。同様に、設備処理機能26もハードコードされ、呼に含まれる物理リソースに関連したすべてのデータ処理を実現する。スイッチ機構28は、Northern Telecom社などのスイッチメーカーから提供される一体式ソフトウェアを実行するスイッチおよびコンピュータのハードウェア構成要素を表す。スイッチ機構28は、接続を確立するために必要な物理的機構を提供し、伝達装置(TIとDSO)、スイッチングマトリクス装置(ネットワーク平面とその処理装置)、リンク層信号処理装置(SS7、MTP、ISD、LAPD)、および専用回路(コンファレンス・ポート、音声トーン検出器)を含むがこれらに限定されない。

#### 【0009】

前述の課題に取り組む試みにおいて、国際電気通信連合と欧州電気通信標準協会は、ITU-Tインテリジェント・ネットワーク規格(Intelligent Network Standard) (「IN」) を承認した。同様に、Bellcoreは、高度インテリジェント・ネットワーク規格(Advanced Intelligent Network Standard) (

(10)

「A I N」) を承認した。これらの2つの規格は、表現と発展の状態が異なるが、ほとんど同じ目的および基本概念を有する。したがって、これらの規格は、サービス処理機能22がスイッチから分離された1つのネットワーク・アーキテクチャと見なされる。

#### 【0010】

I NおよびA I Nアーキテクチャを使用することによって、本質的に所与のタイプの呼において起動されるサービスに依存しない構成単位 (Service Independent Building Blocks) (「S I R S」) のテーブルである新しいサービス論理プログラム (Service Logic Program) (「S L P」) を作成し展開することによって新しいサービスを公開することがおそらく可能である。この手法により、いくつかの特定のエレメント・タイプが、S L Pと相互動作して、ネットワーク加入者にサービスを提供する。その結果、新しいまたは可能性のあるサービスが、既存のS I B B Sによって制限される。

#### 【0011】

全体が40で示されたI NまたはA I Nアーキテクチャは、一体式スイッチ20の機能を、サービス制御ポイント (Service Control Point) (「S C P」) 42と、サービススイッチングポイント (Service Switching Point) (「S S P」) およびスイッチングシステム44に論理的に分離する。S C P 42は、サービス処理機能22を含むが、S S Pおよびスイッチングシステム44は、呼処理機能24、設備処理機能26およびスイッチ機構28を含む。このケースでは、呼処理機能24、設備処理機能26およびスイッチ機構28は、グループ46によって記号化されたようにハードコードされ、混合され、画一化されている。

#### 【0012】

サービススイッチングポイント (「S S P」) は、加入者の信号発信が、ダイヤルされた番号だけに基づいた単純なものでない経路指定を必要とすることを認識するためにスイッチにある機能モジュールである。S S Pは、呼のさらなる操作を中断し同時にリモートS C P 42に呼の適切な処理に関する問い合わせを行い、本質的にいくつかのスイッチのデータベース・サーバとしてはたらく。この

(11)

処理の分割によって、特別なサービス呼を処理するための希であるが時間のかかるタスクの負荷がスイッチから軽減される。さらに、この適度な集中化により、ネットワーク全体のために働く容易に修正可能で負担の重い1つのレポジトリを有することと、すべてのスイッチにあるレポジトリの完全なコピーを展開することとの平衡がとれる。

#### 【0013】

次に図2を参照すると、INまたはAINアーキテクチャを使用する通信システムが示され、全体が50で表される。ISDN端子52、第1の電話54、第2の電話56などの様々な顧客システムが、SSPおよびスイッチングシステム44に接続される。ISDN端子52は信号線60および伝送線62によってSSPおよびスイッチングシステム44に接続される。第1の電話54は、伝送線64によってSSPおよびスイッチングシステム44に接続される。第2の電話56は、伝送線68によってリモートスイッチングシステム66に接続される。また、リモートスイッチングシステム66は、伝送線70によってSSPおよびスイッチングシステム44に接続される。

#### 【0014】

図1に関して前に説明したように、SSP70は、加入者の信号発信が、ダイヤルされた番号にのみ基づいた単純なものではないルーチングを必要とすることを認識するためにスイッチにある機能モジュールである。SSP70は、呼のさらなる処理を中断し、呼の適切な処理のための問い合わせ(query)を行う。この問い合わせは、リモートSCP42にSS7メッセージの形で送られる。サービス制御ポイント42は、その場所にあるデータ・ベース内容を変更することにより、多くの内在するスイッチを介して接続された加入者が使用できるネットワーク機能を変更することができるためそのように名前を持つ。問い合わせは、信号線72を介して、そのようなエレメントの中で単にSS7メッセージ用のルータである信号転送ポイント(Signal Transfer Point) (「STP」) 74に送られ、次に信号線76を介してSCP42に送られる。

#### 【0015】

統合サービス管理システム(Integrated Service Management System) (「I

(12)

SMS」) 78は、サービスを展開または変更したり加入者ごとのサービスへのアクセスを管理したりする管理ツールとして構想されている。ISMS 78は、主に、SSP 70とSCP 42に記憶された演算ロジックとデータを変更することによって動作する。ISMS 78は、様々なユーザ・インタフェース80および82を有する。このISMS 78は、操作線84によってSCP 42に接続され、操作線86によってSSPおよびスイッチングシステム44に接続され、操作線90によってインテリジェント周辺装置(Intelligent Peripheral) (「IP」) 88に接続される。インテリジェント周辺装置88は、音声応答システムや音声認識システムなど、スイッチ上で利用できない機能をネットワークに追加するために使用される装置である。IP 88は、信号線92と伝送線94によってSSPおよびスイッチングシステム44に接続される。

#### 【0016】

次に図2を参照して、従来技術による呼の処理について説明する。顧客が受話器を取ってダイヤルし始めたときに呼が行われる。会社ノスイッチモニターにあるSSP 70が、ダイヤリングを監視し、トリガ・シーケンスを認識する。SSP 70は、サービス・ロジックを調べることができるまで呼のさらなる処理を中断する。次に、SSP 70は、標準的なSS 7メッセージを構成し、それをSTP 74を介してSCP 42に送る。SCP 42は、メッセージを受け取ってデコードし、SLPを起動する。SLIは、番号変換用のデータベース・ルックアップなどの他の機能を作動させることができるSCPを解釈する。SCP 42は、その呼の処理に関するSS 7メッセージをSSPおよびスイッチングシステム44に返すか、あるいは適切なサービスを実行するためにネットワーク・エレメントにメッセージを送り出す。呼の終わりに、呼を切断するためにスイッチの間でSS 7メッセージが送られ、呼に関与する各スイッチが呼の詳細レコードを作成する。呼の詳細レコードは、各呼ごとにオフラインで収集され、関連付けされ、解決され、それにより市外通話の課金が導出され、呼の処理が完了する。

#### 【0017】

INおよびAINアーキテクチャは、予測可能なすべてのサービスをサポートするために1組の標準機能を事前定義しようとする。そのような標準機能は、ス



スイッチ内の様々な状態機械にすべてハードコードされる。残念ながら、新しい技術または予想外のサービス要求と関連して生じる可能性のある新しい機能は、多数のベンダ・プラットフォームに渡るネットワーク・ソフトウェアの広範囲なオーバーホールと試験なしには実現できない。さらに、新しい機能が、標準化された呼出しモデル、プロトコルまたはインタフェースに対する変更を必要とする場合は、その変更が業界標準グループによって承認されるまで、その機能を利用するサービスの実現が遅れることがある。しかし、規格案が、INとAINがサポートする機能セットを拡大しようとしているにもかかわらず、装置の供給業者は、コードの複雑さの驚異的な増大のために、そのような規格案を支持することを拒絶してきた。

#### 【0018】

図2をさらに参照すると、INおよびAINアーキテクチャの他の制限は、スイッチ内で動作する呼処理機能と設備処理機能すなわちSSP70を有することにより生じる。その結果、これらの機能は、独自のソフトウェアを使用して各スイッチメーカーから提供されなければならない。したがって、ネットワーク所有者は、新しい機能をサポートするためにメーカー・ソフトウェア・リリースにまだ大きく依存している。問題をさらに複雑にするのは、ネットワーク所有者が、統合された開発および試験環境において、SSP70モジュールを他のモジュールと一緒に試験できないことである。さらに、スイッチメーカーの処理環境用のSSP70が、ネットワーク所有者のサービス作成環境と互換性をもつという保証はない。

#### 【0019】

このように複数のネットワーク所有者が共通の組のスイッチメーカーに依存することにより、競合を制限する望ましくない2つの状況が生じる。第1に、メーカーのソフトウェア・リリースは、いくつかのネットワーク所有者から要求された変更を組み込もうとすることがあり、それによりネットワーク所有者は、自分たちのサービスを競合により提供されるサービスと真に区別することができない。これにより、メーカーが新しいリリースに他のネットワーク所有者からの要求を組み込むまで、いくつかのネットワーク所有者が待たされる。第2に、あるネ

(14)

ネットワーク所有者から新しいサービスの実装を要求された機能を含むスイッチソフトウェア・リリースは、意図せずに他のネットワーク所有者に利用可能になることがある。したがって、INおよびAINアーキテクチャの意図にもかかわらず、ネットワーク所有者が、ネットワーク・サービスの挙動を形づくる機能要素を完全に制御したり利用したりできないため、ネットワーク所有者の新しいサービスの作成、試験および展開はまだ妨げられる。

#### 【0020】

これらの問題を解決する1つの試みにおいて、全体が150（図1）として示された個別スイッチインテリジェンスおよびスイッチ機構（Separate Switch Intelligence and Switch Fabric）（「SSI/SF」）は、スイッチングシステム44からSSP70を論理的に分離する。次にまた図1を参照すると、スイッチインテリジェンス152は、対応するハードコードされた状態機械エンジンによって個別の状態テーブルにおいてコード化され、円154と156で記号化された呼処理機能24と設備処理機能26を含む。スイッチ機構機能158とスイッチインテリジェンス機能152との接続は、通信ネットワークによって拡張することができ、その結果、スイッチ機構158とスイッチインテリジェンス152は、物理的に一緒に配置されたり、同じプロセッサ内で実行されたり、さらに1対1に対応する必要はない。その結果として、スイッチインテリジェンス152は、すべてのスイッチに共通でサービスに固有でなくまたメーカーに固有でない簡単な機能の整合性のあるインタフェースを提供する。

#### 【0021】

インテリジェント計算複合体（Intelligent Computing Complex）（「ICC」）160は、複数のスイッチインテリジェンス要素152と通信するサービス処理機能22および30を含む。この手法は、ほとんどの基本機能がメーカー固有のコードの領域の外に移動されるため、ネットワーク所有者に、フレキシブルなサービス実装の利点を提供する。サービス・ロジックの作成、開発、試験および実行のために、より統合された環境を提供することによって、さらなる改良を実現することができる。

#### 【0022】

(15)

前に考察したように、現在のネットワークスイッチは、一体式で独自のハードウェアとソフトウェアに基づく。ネットワークスイッチは、数百万ドルのコストがかかることがあるが、そのような設備は、現在利用可能な計算技術の観点から見たときに処理速度が比較的遅い。たとえば、そのようなスイッチは、60 MHzのレンジで動作する縮小命令セット・コンピューティング（「RISC」）に基づき、一般にスイッチング網内の様々なプラットフォーム間で9.6 kb/秒の伝送速度をサポートするX.25などのデータ通信プロトコルを使用して互いに通信する。これは、200 MHz以上で動作するプロセッサを含むパーソナル・コンピュータと、150 mb/秒FDDIおよびATMインタフェースを提供するハイエンド・コンピュータ・ワークステーションと比較してきわめて遅い。したがって、ネットワーク所有者は、独自のハードウェアの代わりにハイエンド・ワークステーションを使用することが必要である。

【0023】

【発明が解決しようとする課題】

本発明は、インテリジェント通信ネットワークの複数の分散サービス・ノードのそれぞれと物理的に関連付けられたスイッチやルータなどのリソース複合体において受け取ったイベントおよびサービス要求をすべて処理するリアルタイムサービスを提供するサービス制御システムを対象とする。

【0024】

【課題を解決するための手段】

一般に、本発明のサービス制御コンポーネントは、サービス要求の処理において、たとえばATMスイッチ、インターネット・ゲートウェイ、インテリジェント周辺装置、他のスイッチまたはルータ・リソースなどのインテリジェント・ネットワーク・リソース複合体に指示することができ、さらに、サービス要求を処理するために必要なインテリジェンスを含む。詳細には、組み込まれたインテリジェンスにより、サービス制御コンポーネントは他のインテリジェント・ネットワーク・コンポーネントと対話して他のロジックにアクセスするか、またはサービス論理インスタンスを処理するのに必要な情報（サービスまたはユーザ・データ）を獲得することができる。サービス制御は、リアルタイムサービス処理にお

(16)

いてリソース複合体およびローカル・データ管理システムと接続し対話し、インテリジェント・ネットワークが提供するサービスの試み进行处理するのに必要な論理および処理能力を保有する。サービス制御は、インテリジェント・ネットワークのサービス管理者とデータ管理コンポーネントによって管理、更新、管理される。インテリジェント・ネットワークは、呼を受け取る呼スイッチングプラットフォームまたはリソース複合体に依存せずまたそれに対して透明なインテリジェント呼処理サービスを提供し、呼イベント进行处理するように容易に適合される。したがって、高価なベンダ固有のハードウェア、オペレーティング・システムおよびスイッチングプラットフォームへの依存がなくなる。分散インテリジェント・ネットワークは、さらに、場所に依存しないイベント処理サービスの実行をサポートし、モジュール式ソフトウェア論理プログラムを、アーキテクチャ内の実質的にどこでも実行することができるようにし、そのような分散プロセス間の場所に依存しない通信を実現し、したがって特別なサービス・ノードが不要になる。

#### 【0025】

より詳細には、本発明は、リソース複合体によってサービス制御コンポーネントにサービス要求が送られたときに開始される1つまたは複数のプロセスを制御する。サービス制御は、他のコンポーネントと対話して、要求サービスを提供するのに必要な要求データにアクセスする。要求サービスの挙動シーケンスが完了するかサービス・ユーザがサービスを使用するのをやめたとき、プロセスは完了する。処理の終わりに、要求の代わりにサービス要求側にサービスを提供することに関わるすべてのリソースが開放される。各サービス要求は、サービス処理の1つのインスタンス（スレッド）を開始し、それにより偶発性またはボトルネックの少ない大量の並列処理が実現される。

#### 【0026】

各サービス・スレッドのインスタンスは、イベントと関連した所定の優先順位に従って記憶し実行するための適切なサービス・スレッド待ち行列への特定の呼インスタンスに関して、それ自体のイベント待ち行列を、受け取った非同期チャネリング・イベントを提供するサービス制御によって維持することが好ましい。

(17)

サービス制御は、さらに、スイッチ／リソース複合体へのイベントの非同期チャネリングを提供し、またはそれと一緒に他の実行サービス論理プログラムを提供し、応答を待つときスレッド・インスタンスは遮られる。

【0027】

本発明によれば、サービス制御コンポーネントの主な役割は、スイッチングプラットフォームやその他の外部リソースからイベントまたは要求を受け取って処理し、受けた要求を処理するためにサービス論理プログラムを識別し起動し、データ管理記憶装置からネットワーク・オペレーティング・システム（NOS）を介して、またはデータ・ベース・アプリケーション・プログラム・インタフェース（API）を介して直接、サービスまたは加入者に関連するデータを要求し、NOSを介してサービスまたは加入者に関連するデータをデータ管理コンポーネントに更新し、優先順位を付けたイベントとメッセージをリソース複合体またはその他の論理プログラムに送ってユーザの対話を制御する機能を提供し、リソース複合体から、たとえばPIN、選択したメニュー項目などに対応するデュアル・トーン・マルチ・フリケンシ）DTMF数字などのユーザ入力を含むメッセージ・セットを受け取り、同じサービス処理インスタンスに含まれるすべての関与物の状況とデータを維持し、課金レコードを生成してそれをデータ管理コンポーネントの課金レコード生成機能に送ることを含む。

【0028】

本発明を特徴づける新規性の様々な特徴は、添付し本開示の一部を構成する特許請求の範囲に詳細に示される。本発明とその動作上の利点およびその使用により達成される特定の目的をより良く理解するために、本発明の好ましい実施形態を例示し説明した図面および説明を参照されたい。

【0029】

本発明の以上その他の利点は、添付図面と共に以下の説明を参照することによってより良く理解することができる。

【0030】

【発明の実施の形態】

本発明は、本明細書においてインテリジェント分散ネットワーク・アーキテク

(18)

チャ (Intelligent Distributed Network Architecture) (「IDNA」) または次世代インテリジェント・ネットワーク (Next Generation Intelligent Network) (「NGIN」) と代わりに呼ばれる包括的インテリジェント・ネットワークの1つのコンポーネントである。本明細書で説明するように、NGINアーキテクチャは、たとえばスイッチ、ルータ、IP終端アドレスなどのリソース複合体またはスイッチングプラットフォームにおいて受け取った任意のタイプの呼にインテリジェント呼処理サービスを実行するように設計される。IDNA/NGINは、複数の分散したサービス・ノードを含み、各ノードは、その特定のサービス・ノードと物理的に関連付けられたスイッチまたはリソース複合体において呼を受け取った場合にその呼を処理するのに必要な呼処理機能を提供する実行環境を提供する。NGINは、きわめてスケーラビリティの高いアーキテクチャであり、独立したサービス論理プログラム (Service Logic Program) (「SLP」) として実施される実行可能なサービス・オブジェクトと、1-800電話呼やファックス送信などのイベント・サービスを実行するための関連データとを、サービス・ノードにおいてコスト効率の高い方法で確実に展開し維持することができるように設計される。CORBA準拠のオブジェクト・リクエスト・ブローカー技術を利用することによって、インテリジェント・ネットワークは、イベントまたは呼を受け取るイベントスイッチングプラットフォームまたはリソース複合体から独立しそれに対して透明な場所とプラットフォームに依存しない呼処理サービスの実行をサポートし、サービス実行プラットフォームに関係なしに事実上ネットワーク上のどこでも高レベルの論理プログラムを実行することを可能にする。さらに、システムは、そのような分散プロセス間で場所に依存しない通信を提供する。

#### 【0031】

次に図1を参照すると、インテリジェント分散ネットワーク・アーキテクチャ (Intelligent Distributed Network Architecture) (「IDNA」) は、一般に、170として示される。本発明は、SSI/SFアーキテクチャ150のICC160とスイッチインテリジェンス152をインテリジェント呼処理装置 (「ICP」) 172に統合する。状態テーブルによって機能が定義されたSSI

／SFアーキテクチャ40のINまたはAINと違い、ICP172は、ブロック174、176および178によって記号化されたオブジェクト指向プラットフォームにおける管理対象オブジェクトとして、サービス制御機能22、呼処理機能24および設備処理機能26を含む。ICP172は、リソース複合体180から論理的に分離される。

#### 【0032】

次に図3を参照して、本発明によるインテリジェント分散ネットワーク・アーキテクチャを利用する通信システムが示され、全体が200として示される。広域ネットワーク（「WAN」）202は、広い地理的領域の全体にわたってアプリケーションとデータの配布をサポートするシステムである。転送ネットワークは、光同期伝達網（Synchronous Optical NETwork）（「SONET」）に基づいて、IDNAノード204を接続し、それらのノード内のアプリケーションが互いに通信できるようにする。

#### 【0033】

各IDNAノード204は、インテリジェント呼処理装置（「ICP」）172とリソース複合体180を含む（図1）。図3は、リソース複合体A（「RCA」）206とリソース複合体B（「RCB」）208を有するIDNAノード204を示す。ICPは、設備提供、課金、復旧などの既存の支援機能を提供する補助処理装置210に連結することができるが、それらの機能は、ネットワーク管理システム（「NMS」）212によって提供される機能に吸収されてもよい。しかしながら、好ましい実施形態において、図4（a）と関連して本明細書で説明するように、それらの支援機能は、データ管理（Data Management）（「DM」）コンポーネント400を有する集中型のサービス管理（Service Administration）（「SA」）システム500によって提供することができる。図3にさらに詳細に示したように、ICP172は、シグナリング216と伝達リンク214を有する直接リンク214によって、他のICP172、他のネットワーク（図示せず）、または他の装置（図示せず）に連結することができる。直接リンクは、接続された装置の間の待ち時間をなくし、装置が自己の言語で通信することを可能にする。ICP172は、IDNAノード204の「頭脳」であり、

(20)

I DNA ノード 2 0 4 の処理要件によって、1つの記憶装置を備えた1つの処理装置から大規模コンピュータ・ネットワークにわたることができる汎用コンピュータであることが好ましい。汎用コンピュータは、冗長処理、メモリ記憶および接続を有する。

#### 【0034】

汎用コンピュータは、本明細書で使用されるとき、電話のスイッチング用に特別に構成され設計された専用の装置と対照的な商用既製コンポーネントまたはそれにより組み立てることができるコンピュータのことを指す。呼ネットワーク内の汎用コンピュータの統合は、多数の利点を提供する。

#### 【0035】

汎用コンピュータを使用することにより、I C P 1 7 2 は、高い処理要求を満たすために付加的なハードウェアで拡張する機能が与えられる。そのような付加物は、処理能力、データ記憶および通信帯域幅を高める機能を含む。そのような付加物は、呼ネットワーク内の各スイッチのメーカー固有のソフトウェアおよび／またはハードウェアの変更を必要としない。したがって、スイッチング網内の個々の装置を修正することなく新しいサービスとプロトコルを全般的なスケールで実装しインストールすることができる。一体式スイッチ 2 0 (図 1) からインテリジェント呼処理装置 1 7 2 への変化は、本発明に、以上の利点と高い機能を提供する。

#### 【0036】

より高い処理能力を必要とする用途の場合には、マルチプロセッシングによって、より安価なプロセッサを使用して呼処理の価格／性能比を最適化することができる。他の用途では、より高い処理速度を有するミニコンピュータなどのより高性能な装置を使用することが有利であったり必要であったりコスト効率が高かったりすることがある。

#### 【0037】

前に述べたように、I C P 1 7 2 は、たとえば U N I X (登録商標) または W i n d o w s N T (登録商標) オペレーティング・システム上で動作する汎用コンピュータのクラスタを含む。たとえば、1つのリソース複合体で最大 1 0 0



(21)

、000ポートをサポートする大きなアプリケーションでは、ICP172は、対称型マルチプロセッサ・クラスタにおいて333MHzで動作する16個の32ビット・プロセッサからなる。たとえば、プロセッサを、それぞれ4つのプロセッサを備えた別個の4つのサーバに分割することができる。個々のプロセッサは、システム・エリア・ネットワーク (System Area Network) (「SAN」) やその他のクラスタ化技術によって接続される。プロセッサ・クラスタは、Redundant Array of Independent Disks (「RAID」) モジュール式データ記憶装置へのアクセスを共用することができる。共用される記憶装置は、モジュール式ディスク装置を追加したり除去することによって調整することができる。クラスタ内のサーバは、RC180 (図1) に対する冗長リンクを共用することが好ましい。

#### 【0038】

図に示しパーソナル・コンピュータの「プラグ・アンド・プレイ」機能と同じように、ICPソフトウェア・アーキテクチャは、(1) 管理ソフトウェア、(2) ICPアプリケーション、(3) 計算ハードウェアおよびソフトウェア、(4) リソース複合体コンポーネント、ならびに(5) サービス・アーキテクチャおよび処理の互換性を可能にするオープン処理モデルである。そのような汎用アーキテクチャは、標準化により保守コストを削減し、量産効果によって得られる利益を提供する。

#### 【0039】

したがって、本発明は、開発作業の分割とモジュール式ツールの使用を可能にし、それによりサービスの開発と実装が高速化する。さらに、サービス管理の使用と関連する側面は、固定されたメッセージング・プロトコルまたは所与のメーカーから供給されるハードウェアとソフトウェアの特定の組合せにより受ける制約と対照的に、必要な基準に基づくネットワーク・オペレータの制御の範囲内にある。

#### 【0040】

管理対象オブジェクトの使用により、本発明は、また、容量や使用量などの任意数の要因に基づいて、ネットワークの全体にサービスと機能をフレキシブル (

(22)

「望む場所に」) かつ動的に(「実行中に」) 分散させることができる。サービス処理22(図1)、呼出し処理24(図1)および設備処理26(図1)が同種のプラットフォームで動作するため、性能が向上する。さらに、本発明は、以前に利用できなかった呼サブエレメントの監視と処理を可能にする。本発明は、また、機能とサービスを監視し、それらが旧式になったり使用しなくなったりしたときにそれらを削除することができる。

#### 【0041】

リソース複合体(System Area Network) (「RC」) 180(図1)は、伝送、シグナリングおよび接続サービスを提供する物理装置またはリソースの集合である。インテリジェント周辺装置88を含むことができるRC180は、INまたはAINあるいはSSI/SFアーキテクチャのスイッチ機構28および158(図1)と置き換わる。INまたはAINアーキテクチャと違って、RCA206などのリソース複合体の制御は低レベルである。さらに、RCA206は、複数のスイッチ機構158を含むことができる。スイッチ機構158や他の顧客インタフェース(図示せず)は、標準的な電話接続によって複数の加入者およびスイッチング網に接続する。そのような顧客システムは、ISDN端末52、ファックス装置220、電話54およびPBXシステム222を含むことができる。ICP172は、高速データ通信パイプ(最低100mb/秒のイーサネット(登録商標)接続)224を介してRC180(図1)、RCA206およびRCB208と通信する。RC180、206および208は、プリンタと類比することができ、ICP172は、パーソナル・コンピュータに類比することができ、ここでパーソナル・コンピュータは、ドライバを使用してプリンタを制御する。IDNAノード204内の「ドライバ」は、図5を参照して後で説明するリソース複合体プロキシ(Resource Complex Proxy) (「RCP」) (図示せず)である。これにより、メーカーは、IDNAモデルを組み込むためにすべてのソフトウェアを書き換えることなく、このインタフェースを使用してIDNA準拠ノードを提供することができる。

#### 【0042】

さらに、リソース複合体180(図1)、RCA206およびRCB208の

制御は、一般にA I NまたはI Nアーキテクチャによって提供されるものよりも低いレベルにある。その結果、リソース複合体メーカーは、設備およびネットワーク管理処理をサポートするために1つのインタフェースを提供するだけでよく、特定の呼およびサービスの処理をネットワーク所有者に提供する必要がない。低レベルのインタフェースが、より個別の操作に抽出される。インタフェースが1つであるため、ネットワーク所有者は、価格と性能に関する決定に基づいて、広い範囲のリソース複合体から選択することができる。インテリジェンスは、R C 1 8 0ではなくI C P 1 7 2に追加され、R C 1 8 0が変更から分離され、その複雑さが減少する。R C 1 8 0の役割が簡素化されるため、変更がより容易に行なわれ、それにより、非同期転送モード (Asynchronous Transfer Mode) (「A T M」) などの代替のスイッチングおよび伝送技術への移行が容易になる。

#### 【0 0 4 3】

インテリジェント周辺装置 (Intelligent Peripheral) (「I P」) 8 8は、実際の呼伝送経路に含まれる情報を処理しそれに基づいて動作する機能を備える。I P 8 8は、一般に、R C B 2 0 8などの個別のリソース複合体の中にあり、R C A 2 0 6と同じようにI C P 1 7 2によって制御される。I Pは、デジタル信号処理 (Digital Signal Processing) (「D S P」) 技術を使用してリアルタイムで呼伝送経路内のデータを処理する機能を提供することができる。

#### 【0 0 4 4】

ネットワーク管理システム (Network Management System) (「N M D」) 2 1 2は、I D N Aネットワーク2 0 0内のハードウェアとサービスを監視し制御するために使用される。提案されるN M S 2 1 2の実施態様は、I D N Aネットワーク2 0 0内のコンポーネントの管理を行う電気通信管理網 (Telecommunications Management Network) (「T M N」) 準拠のフレームワークでよい。より具体的には、ネットワーク管理システム2 1 2は、サービスの展開を制御し、それらのサービスの健康を維持し、それらのサービスに関する情報を提供し、I D N Aネットワーク2 0 0のネットワーク・レベル管理機能を提供する。N M S 2 1 2は、I D N Aノード2 0 4内のエージェント機能によりサービスおよびハードウェアにアクセスしてそれを制御する。I D N Aノード2 0 4内のI C P - N

(24)

MSエージェント（図示せず）は、NMS 212から出されたコマンドまたは要求を実行する。NMS 212は、標準の操作リンク 226によってRCA 206とRCB 208を直接監視し制御する。

#### 【0045】

さらに図3に示したように、管理対象オブジェクト環境（Managed Object Creation Environment）（「MOCE」）228は、IDNAネットワーク200内で実行するサービスを作成するサブコンポーネントを含む。サービス設計者が新しいサービスを作成するために使用するサービスに依存しない構成単位およびAPI表現は、MOCEの主要サブコンポーネントであるグラフィック・ユーザ・インタフェース（「GUI」）に埋め込まれる。MOCE 228は、サービス作成（Service Creation）（「SC」）環境とも呼ばれる1つのユーザ環境またはプラットフォームのホストとして働くツールの統合された集まりである。これは、管理対象オブジェクトおよびサービス試験に含まれるサービス・ドキュメンテーション、管理対象オブジェクトの定義、インタフェースの定義、プロトコルの定義、データ入力の定義などのサービス作成のプロセス全体に必要とされる処理の集まりである。管理対象オブジェクトは、ネットワーク上のすべてのノードに適用することができるため、ネットワーク所有者は、MOCE 228を使用して1度に1つのサービスを開発するだけでよい。これは、様々なスイッチメーカーのそれぞれにそのメーカーのバージョンのサービスを開発させ、それによりサービスを何度も開発しなければならないネットワーク所有者と対照的である。

#### 【0046】

MOCE 228とNMS 212は、レポジトリ230を介して接続される。レポジトリ230は、NMS 212によって配布されIDNA/NGINノード204内で使用される管理対象オブジェクトを含む。レポジトリ230は、また、MOCE 228とNMS 212の間のバッファを提供するが、MOCE 228は、点線232で示された「生の」ネットワーク試験を実行するためにNMS 212に直接接続されてもよい。

#### 【0047】

図4（a）に示したような本発明の好ましい実施形態によれば、IDNA/N

(25)

GINシステムは、追加された機能と共に、記憶（レポジトリ）230機能とIDNAシステム170の汎用ネットワーク管理（NMS）212機能の両方を提供する集中されたサービス管理（Service Administration）（「SA」）コンポーネント500を含む。一般に、図4（a）に示したようなSAコンポーネント500は、IDNA/NGINシステムのすべてのサービスとデータのオフライン記憶、命名、配布、活動化および除去をサポートし、さらに、IDNA/NGINサービス・ノード内のサービス・オブジェクトによって使用されるデータの実行時記憶、複製、同期および利用を可能にするデータ管理（「DM」）機能を提供する。

#### 【0048】

より詳細には、図4（b）に概念的に示したように、サービス管理コンポーネント500は、IDNAサービス処理ノードによって使用されるすべてのサービスとサービス・データを管理、記憶、配布し、システムIDNA/NGINに実装されたハードウェア・コンポーネントとソフトウェア・コンポーネントの両方を構成するために必要なすべての機能を実行するコンポーネントである。一般に、図4（b）において示したように、SAコンポーネント500の役割は、MOCE（サービス作成）228からデータを受け取り、受注やその他のレガシ・システム229から顧客注文データ502を受け取り顧客の使用のためにIDNA/NGINシステムに供給し、たとえばサービス作成プロセス中に、たとえばMOCE/SCEユーザによる要求に応じて、データ、サービスに依存しない構成単位（Service Independent Building Block）（「SIBB」）、サービス論理プログラム（Service Logic Program）（「SLP」）および他のサービス・ロジック・コンポーネント503を、たとえばMOCE200に展開し、MOCE228から、完成し試験済みのサービス・パッケージ、SIBB、SLPその他のサービス・ロジックまたはデータ・コンポーネント506を受け取り、各サービス・コンポーネントに独特な名前を提供し、本明細書において後でより詳細に説明するように、データと各サービス・コンポーネント509をデータ管理機能コンポーネント600に配布することである。さらに、図4（a）に示したように、サービス管理300は、データ管理コンポーネント600がそのデータをす

(26)

すべての受け取るすべてのIDNAサービスとデータを含むグローバルなレコード・データベース(Database of Record) (「DBOR」)を含むレポジトリ230を維持する。

#### 【0049】

サービス管理のその他の役割は、データおよびサービス・コンポーネント512を活動化させて、データ、SIBBおよび管理対象オブジェクトまたはサービス論理プログラムSLPが、データ管理コンポーネント600を介してノードに利用可能になることを保証し、後で詳細に説明するように、データ、SLPおよびSIBB515の名前を、登録のために、その論理名をネットワーク・オペレーティング・システム(「NOS」)コンポーネント700に送ることによって登録し、データおよびサービス・コンポーネント518を停止させ、データ管理コンポーネント600を介してIDNA/NGINシステムからデータとサービス521を取り除くことを含む。サービス管理は、さらに、命名プロセスによる改訂の他に、それぞれのSIBBおよびサービスの状態(前試験済み、後試験済み、展開済みなど)を維持することによってコンフィギュレーション管理機能を実行する。これにより、そのサービスのすべてのコンポーネントを首尾良く試験し構成できるまでサービスが展開されないことが保証される。

#### 【0050】

図4(b)にさらに詳しく示したように、サービス管理コンポーネント500は、さらに、SAが受け取るコンフィギュレーション情報にしたがってIDNA/NGINサービス・ノード204を構成し供給する機能を実行する。詳細には、受け取ったコンフィギュレーション情報に基づいて、SAコンポーネント500は、各サービス・ノード204hにおける各コンポーネントの機能、どのサービスとデータをどのノードに配布するか、サービス・ノードにあるどのサーバでサービスを行うか、およびIDNA/NGINノード・サーバと関連したローカル・メモリ・レジデントにどのデータをキャッシュするかを決定する。詳細には、SAは、各サービス・ノードに配置されたローカルLENキャッシュに記憶するために、サービス・プロファイル(コンフィギュレーション)ファイル580に含まれるコンフィギュレーション規則を、NOSシステム700のローカル(ノード)

リソース管理（「LRM」）コンポーネント575に展開する。そのようなコンフィグレーション・ファイル580は、IDNAノードで実行するサービスを決定する。LENは、最初に、そのノードのローカル・キャッシュに記憶されたこのサービス・プロファイル・ファイル580を読み取り、サービス・プロファイル・ファイル内の規則に従ってサービスを実行する特定のサービス層実行環境（Service Layer Execution Environment）（「SLEE」）たとえば仮想機械を決定し、SLEEにおいて（永続的なオブジェクトとして）活動的に実行するかまたは要求時のみインスタンス生成するサービスを決定する。

#### 【0051】

図4（a）を再び参照すると、NGINデータ管理コンポーネント600は、サービスのライフサイクルとサービスの利用機能力の両方において機能する。サービス管理コンポーネントが、グローバルなレコード・データ・ベース（レポジトリ）を維持す場合は、データ管理コンポーネント600は、それぞれのIDNA/NGINサービス・ノードにローカル・データ記憶とデータ管理機能を提供する。これは、サービス・プログラムおよびSIBB、サービス用のデータ（顧客プロファイル、電話番号など）、マルチメディア・ファイル（対話型音声応答（Interactive Voice Response）（「IVR」）用の音声ファイルなど）などを含むすべてのタイプのデータを含む。具体的には、サービス・ノードのデータ管理コンポーネント600は、サービス管理によって指定されたようなローカルNGINサービスによって実行されるサービスに必要なすべてのデータを含むSAGlobalDBORの抽出物を受け取る。このメカニズムは、図4（c）と関連して後でより詳細に説明する。

#### 【0052】

好ましい実施形態において、SAコンポーネントのデータ管理コンポーネント600は、各IDNA/NGINサービス・ノードにローカル・データ記憶および管理機能を提供する。詳細には、データ管理は、サービス管理から受け取ったデータを1つまたは複数のデータ・ベースに記憶し、必要なデータをサービス制御コンピュータ内のメモリ・レジデントまたは同じ場所に配置されたデータベース・サーバ上にキャッシュすることによってサービス/データをサービス制御環

境に容易に使用可能にし、それにより、サービス／データを最小の待ち時間でサービス制御サービスに提供することができる。より一般的には、データ管理コンポーネント600は、サービス管理から受け取ったかサービス処理の結果受け取ったかに関係なく、リアルタイム記憶、複製、同期、およびデータの利用を実行する。そのようなDM機能は、1) データ・レポジトリ機能、2) データ操作機能、3) データ利用機能、および4) 課金レコード生成機能としてさらに分類することができる。

#### 【0053】

次に図5を参照して、本発明によるインテリジェント分散ネットワーク・アーキテクチャ200を使用する通信システムの論理機能図を説明する。ICP172は、管理対象オブジェクト基本クラス244から得られた様々な管理対象オブジェクト246、248、250および252をホストとして処理するICP-NMSエージェント240とSLEE242を含むように示される。

#### 【0054】

一般に、管理対象オブジェクトは、ソフトウェア機能を実装するメソッドであり、各管理対象オブジェクトは、管理対象オブジェクトの機能を実現するために機能インタフェースと管理インタフェースの両方を提供する。管理インタフェースは、管理対象オブジェクト機能を利用することができる人と物へのアクセスを制御する。本発明において、IDNA/NGINノード204によって実行されるインフラストラクチャ・ソフトウェアを除くすべての電話用アプリケーション・ソフトウェアは、管理対象オブジェクトと支援ライブラリとして展開される。これは、IDNAノード・ソフトウェアを制御し管理するための統一されたインタフェースと実装を提供する。

#### 【0055】

ノードによって処理される伝送トラヒックを接続し経路指定し終端するネットワーク・エレメントの集合は、集合的にリソース複合体(「RC」)と呼ばれる。SLEE上で実行されるサービス処理アプリケーションは、RC180への制御インタフェースとしてリソース・プロキシ(Resource Proxy)(「RCP」)244を使用する。RCP244はデバイス・ドライバに連結され、SLEE内



のオブジェクトからの装置に依存しないコマンドをRC180によって実行される設備固有のコマンドに適用することができる。RCP224は、RCP244内の資産のベンダ間に共通の基本コマンドを実現するインタフェースとして説明することができる。RCP244は、図示したように、IDNAノード204上で動作する1つまたは複数の管理対象オブジェクトとして実現することができる。代替として、この機能をRC180の一部として提供することができる。NMS212、レポジトリ230およびMOCE228は、図3～図5(a)の考察でそれらの要素の説明と一致している。

#### 【0056】

操作リンク226がNMS212をRC180に直接接続することに注意されたい。これは、ネットワーク・ハードウェアの動作状況を監視する際のネットワーク管理システムのより伝統的な役割に対応する。これは、IDNAアーキテクチャと独立に行うことができる（たとえば、周知のTMN手法を使用して）。さらに、RC180を他のリソース複合体254に接続することができる。また、SS7などの信号216が呼処理環境に直接入ることができるようにICP172に入る直接信号リンク214が示される。ネットワーク周囲で信号を遮ることによって、SS7メッセージは、RC180を通ることなくICE172に直接進むことができる。これにより、信号経路が短くなることによって待ち時間が減少し頑強性が向上する。これに伴う伝送線218が、RC180に接続する。

#### 【0057】

図6は、ICP172内の機能インタフェースの階層化を示す。MOCE228は、管理対象オブジェクト・ソフトウェアとその従属物を生成するシステムである。NMS212は、ICP-NMSエージェント240と呼ばれるICP172内に提供されるエージェント機能に接続することによってICP172の実行を制御する。NMS212は、ICE172上のローカル・オペレーティング・システム(Local Operating System)（「LOS」）260の動作を制御する。NMS212は、プロセスの開始と停止、プロセス・テーブルの内容およびプロセスの状況の問い合わせ、オペレーティング・システム・パラメータの構成、およびICP172のホストとして働く汎用コンピュータ・システムのパフォー

マンスの監視を含む I C E 1 7 2 の動作を制御する。

【0058】

NMS 2 1 2 は、また、広域ネットワーク・オペレーティング・システム (Wide Area Network Operating System) (「WANOS」) 2 6 2 の動作を制御する。NMS 2 1 2 は、WANOS 支援プロセスの初期設定と動作および WANO S ライブラリの構成を、LOS 2 6 0 のその制御ならびに NMS S L E E 制御によって提供される他のインタフェースを介して制御する。NMS 2 1 2 は、I C E 1 7 2 上で動作する1つまたは複数の S L E E 2 4 2 のインスタンス生成と動作を制御する。LOS 2 6 0 は、汎用コンピュータを操作するための市販の既製オペレーティング・システムである。WANOS 2 6 2 は、計算ノード間のシームレスな通信を容易にする市販の既製ミドルウェア・ソフトウェア・パッケージ (たとえば、オブジェクト・リクエスト・ブローカー) である。S L E E 2 4 2 は、サービス処理アーキテクチャを実現するソフトウェア・インスタンスである管理対象オブジェクト 2 4 4 の実行のホストとして働く。S L E E 2 4 2 は、I C P - NMS エージェント 2 4 0 による管理対象オブジェクト 2 4 4 の実行を制御する手段を実現する。したがって、S L E E 2 4 2 インスタンスは、管理対象オブジェクト・ソフトウェアの展開と削除、管理対象インスタンスのインスタンス化と抹消、管理対象オブジェクトの対話と協力の支援、固有ライブラリ (Native Library) 2 6 4 に対するアクセスの管理、および必要な制御を実現する際の NMS - I C P エージェント 2 4 0 との対話を可能にするソフトウェア・プロセスである。

【0059】

固有ライブラリ 2 6 4 は、LOS 2 6 0 または WANOS 2 6 2 と固有の汎用コンピュータ実行 (たとえば、コンパイル済みの C ライブラリ) のみに依存するようにコード化されたライブラリである。これらは、主に、S L E E 2 4 2 によって提供される固有の機能を補うために使用される。

【0060】

S L E E ライブラリ 2 6 6 は S L E E 2 4 2 において実行するためにコード化されたライブラリである。それらは、S L E E 2 4 2 および固有ライブラリ 2 6

4によって提供される機能をアクセスすることができる。管理対象オブジェクト244は、SLEE242によってロードして実行されるソフトウェアである。それらは、SLEE242とSLEEライブラリ266（および恐らく在来のライブラリ264）によって提供される機能を利用することができる。

#### 【0061】

ICP-NMSエージェント240は、NMS212にICE172の操作を制御する能力を提供する。ICP-NMSエージェント240は、LOS260の動作と構成、WANOS262の動作と構成、およびSLEE242のインスタンス化と動作を制御する能力を実現する。提案されたサービス処理アーキテクチャは、高抽象化層において動作する。しかしながら、SLEE242の斜視図から、NMS212の制御下で対話するオブジェクト層（ソフトウェア・インスタンス）である管理対象オブジェクト層244と、管理対象オブジェクト242またはSLEE242自体の動作に補足機能を提供するソフトウェア層（SLEE242とLOS260のどちらかに固有）であるライブラリ層264または266の2つの層しかないことが分かる。しかしながら、ある時点で、NMS212が、管理対象オブジェクト・インスタンスの正確な場所の制御を放棄する可能性があることが予想される。たとえば、管理対象オブジェクト・インスタンスは、要求に応じるような1つまたは複数のアルゴリズムまたはイベントに基づいて、あるノードから別のノードに移動することができる。

#### 【0062】

集散的に、LOSおよびWANOSの機能は、図6に示したように、IDNA/NGINシステム・コンポーネント間にプラットフォームに依存せず場所に依存しない接続機能を提供する働きをするネットワーク・オペレーティング・システムまたは「NOS」として表すことができることを理解されたい。すなわち、NOSは、他のIDNA/NGIN機能コンポーネントとサブコンポーネント間のプロセス・インタフェースと通信を提供する1組のネットワーク全体にわたるサービスを含む。NOSによって提供されるサービスの間には、オブジェクト接続性、論理名変換、プロセス間通信、ならびにローカルおよびシステム・ワイド・リソース管理（「RM」）がある。たとえば、図4（a）に示したように、N

OSコンポーネント700は、ローカル(NODE RM)およびシステム・ワイド・リソース管理(SYS EM)機能を提供する。詳細には、NOSコンポーネントは、サービスとデータを必要とするプロセスからサービスの場所をカプセル化し、したがってプロセスは、1つの論理名を呼び出すだけでよい。次に、NOSコンポーネントは、サービスのどのインスタンスを使用するかを決定し、そのインスタンスに接続性を提供する。NOS700は、部分的に、IDNA/NGINの広く分散した性質と、IDNA/NGINのプラットフォーム独立性の両方を使用可能にする。たとえば、前述の論理プログラムは、NOSコンポーネント700を使用して他の論理プログラムを呼び出し、その結果同じサービス・ノードまたはリモート・サービス・ノードの異なるSLEE上で動作する他の論理プログラムを呼び出して起動することができる。詳細には、SA500により、あるサービスだけを実行するようにサービス・ノードを指定することができる。たとえばカンファレンス・ブリッジの接続のような必要なサービスを実行できない関連サービス・ノード204を有するスイッチに呼が到着したとき、IDNAは、そのようなサービスを提供するように構成された別のノードに呼を送らなければならないことがある。IDNAは、NOSコンポーネント700を介して、別のリモート・サービス・ノードにある必要なサービスを呼び出し、呼処理を実行し、元のノードにあるスイッチにサービス応答を提供することが好ましい。

#### 【0063】

次に、図7を参照して、本発明によって管理されたオブジェクトのクラス階層(ヒエラルキー)を説明する。アブストラクトベースクラス管理オブジェクト244は共通の機能性および派生されたすべてのクラスがSLEE242においてオブジェクトとして適切に支持され得ることを保証するためにバーチャル機能を含んでいる。とくに、4つの異なったサブクラス、すなわち、サービス制御クラス252、コール制御クラス250、支持体制御クラス248、オブジェクトリソースプロキシクラス246が示されている。

#### 【0064】

サービス制御クラス252はすべてのサービス機能オブジェクト用のベースク

ラスである。セッションマネージャークラス280はセッション関連情報および活動をカプセル化している。セッションはネットワーク機能の1またはそれ以上のコールまたは他の呼びかけからなることもできる。セッションマネージャークラス280は各セッションに対して唯一の識別子を設けている。コール処理が交点方法で行われるならば、その場合に請求書作成（ビルディング）情報が照合されねばならない。各コール用の唯一の識別子は、コスト的に高い照合処理を必要とするのに代わって、照合を容易にする。サービス処理において、プロトコルは連続する抽象概念層によって包まれている。結局、プロトコルはセッションマネージャの割り当て／例示を正当化するように十分に抽象される（例えば、SS7において、IAMメッセージの受信はセッションマネージメントを有して正当化する）。

#### 【0065】

支持体能力クラス282は支持体上のサービスの品質を変化する。サービス制御クラス252はコールのサービスの品質（“QoS”）の変化を可能にするかまたは同様に56Kbit/sからより高い割合への移動かつ次いで降下のごとき、支持体能力を変化することができる。QoSは接続マネージャークラス302によって管理される。例えば、ハーフレートサブクラス284はコールのQoSを、通常の8KHzのサンプルレートに代えて、4KHzのサンプルレートに低下させる。ステレオサブクラス286は左方チャンネルおよび右方チャンネルを支持するために使用者にコールの2つの接続の形成を許容するかも知れない。

#### 【0066】

サービスアービトレーションクラス288はサービスの対立およびサービスの相互作用の調停を集大成する。これは、サービス制御クラス252が、とくに開始および終了サービスで対立し得るため必要とされる。多くの実用的な理由のために、各サービス制御クラス252内にサービス制御クラス252の他の型との対立を如何に解決するか認識を符号化するのは望ましくない。代わりに、対立が識別されるとき、対立するサービスおよびそれらの未決の要求に対する参照がサービスアービトレーションクラス288に通される。サービスアービトレーションクラス288は、その場合に、多分、局所的な背景、形状データ、および対

立するサービスに対す次の質問を考慮して、適切な行動コースを決定することができる。サービスアービトレーションクラス 288 を有することは、ハードコード化または絶対の機構に対向されるような、明白な文書化および対立解決のアルゴリズムの符号化を許容する。そのうえ、サービスが更新されるかまたは追加されるとき、現存するサービスは、単一のサービス内で多数の関係の変化を必要とし得るあらゆる対立の変化を明らかにするように更新されなくてもよい。

#### 【0067】

特徴クラス 290 は電話による通信と連絡する能力の標準セット（例えば、3 方向呼び出し、キャッチホン）を実行する。このような 1 つの能力は予定された受信者に到達するために現存するコールを接続解除の発生を可能にするオーバーライド 292 にすることができる。他の共通の能力はコールブロック 294 を含むことができ、それにより開始申込みが開始についての 1 組の規準に基づいて拒絶され得る。

#### 【0068】

サービス識別クラス 296 はコール処理の間中他のサービスを選択的に引き起こしかつサービスそれ自体として下位に分類される。サービス識別クラス 296 は柔軟な、背景感知サービス活動を備えかつサービスを活動させるときに決定するための各サービス内に固定したコードを有する必要を除去する。活動シーケンスはサービスそれ自体から隔離されている。例えば、加入者 A および加入者 B は同一組の特徴にアクセスし得る。加入者 A は特定の組の信号を使用して彼のサービスの 1 またはそれ以上を選択的に呼び出すように選ぶ。加入者 B は彼のサービスを活動させるために異なる組の信号を使用するのが好ましい。加入者間の唯一の差異は彼らが彼らのサービスを活動させる方法である。そこで、選択方法をサービスそれ自体から仕切るのが望ましい。利用可能な 2 つの解決がある。加入者 A および B のサービス選択方法が別個のサービス識別クラス 296 において符号化され得るか、またはサービス識別クラス 296 が適切な情報を指示するために加入者当たりの輪郭を使用することができる。これはそのサービスセットが解体されるより多くの使用者に適用されるように導き出され得る。さらに、サービス識別クラス 296 の使用は付与されたコールの背景または進展に基づいてサービ

スへのアクセスのマッピングを変更することを可能にする。このクラスの実行は、多分種々の活動入力を使用して種々のサービスを多数のコール関係者に利用させ得る。従来技術においては、すべてのスイッチ売り主が、この能力を阻止した、柔軟性のないサービス選択スキームを供給した。

#### 【0069】

メディア独立サービスクラス298は、音声、フックス、eメールおよびその他のを含んでいる種々のメディアの型に適用される、記憶および送信300、放送、方向変換、先取り（プリエンプション）、QoSおよび多者接続のごとき、サービス制御クラス252の1型式である。サービス制御クラス252が各メディア型式に適用され得るように開発されるならば、その場合に、サービス制御クラス252は再使用可能なサービス制御クラス252に押し進められ得る。サービス制御クラス252がメディア依存機能およびメディア独立機能（すなわち、サービスおよびセットメディア依存ラッパーSC—メディア型式当たり1つ—を実行するメディア独立機能SC）に押し進められるならば、メディア独立クラス298から派生されるように、記憶および送信300は幾つかのメディア型式のメッセージまたはデータの流れを記憶するような包括的な能力かつ次いでそれを幾つかのイベントに基づいてメディア型式に供給するような能力を備えている。方向変換は特定された条件に基づいて1つの論理アドレスから他の論理アドレスへ接続を移動するような能力を備えている。この概念はコール送信（全ての型式）、ACD/UCD, WATS（1～800サービス）、ファインドミー/フォローミーおよび自動車電話等の基礎である。交渉されるかまたは他の方法で、先取りは、キャッチホン、優先先取り等のごときサービスを含んでいる。QoS変調接続は、音声/フックス、流れているビデオおよびファイル転送のごときパケットネットワークにわたってさらに他のサービスを実行する。多者接続は3方向およびN方向ビデオ会議等を含んでいる。使用者が制御しかつ入力電話のキーを使用して主として実行されるけれども、音声認識が将来において使用者制御および入力に使用されるように期待されている。

#### 【0070】

接続マネージャークラス302はコールに伴われる種々の支持体制御248の

(36)

接続をコーディネートおよび調停する責任を負っている。したがって、多数コール中の参加者間の接続性を管理する複雑さは他のすべてのサービスから分離されかつ取り除かれる。サービスおよびコール処理は接続から結合解除される。このことは1 接続対多数の接続のように接続へのマッピングコールのパラダイムを破壊する。今や、コール対コールのマッピングは多数対多数である。

#### 【0071】

アーキテクチャ内の接続マネージャークラス302は単独で（スタンドアローン）作動するようにまたは対等者として協力するように設計されている。作動において、サービス制御クラス252はコールセグメントを追加、変更および除去するような要求により接続マネージャークラス302を示している。接続マネージャークラス302の義務はこれらの変化をなし遂げることである。留意することは、接続がそれら自体のリソースとしてまたはリソースの属性として検討され得るので、接続マネージャークラス302は基本のリソース管理機能のプロキシまたは態様として実行され得る。

#### 【0072】

コール制御クラス250は電話による通信に共通して使用される基本的な有限状態機のごとき必須のコール処理を実行し、かつコール処理がどのように行われるかを特定する。2つのクラスが開始（コールを配置）304および終了（コールを受容）306の機能的仕切りに沿って導き出され得る。

#### 【0073】

支持体制御クラス248はリソース複合体180へのかつそれからの特定の信号およびイベントを、リソースプロキシ246を経由して、コール制御オブジェクト250によって理解され得る共通の信号およびイベントに適合させるのに向けられている。このクラスから導き出されたオブジェクトの1つの予想される役割は、加入者ライン番号、サービスのクラス、アクセスの型式等のごとき、コールの開始端についての情報を集めることにある。サブクラスが信号送信と関係する回路またはチャンネルの数を基礎にして分化され得る。これらはISDN一次インターフェイス310中の23支持体チャンネル当たり単一信号送信チャンネルに適用されるような、チャンネル関連クラス308、単一回路を制御するのに



ダイヤルするのに使用するアナログ電話 3 1 4 によって類型化されるようなチャンネル単一クラス 3 1 2、および支持体チャンネルから完全に分離された S S 7 信号送信 3 1 8 によって示される、チャンネル共通クラス 3 1 6 を含むことができる。

#### 【0074】

リソースプロキシクラス 2 4 6 は支持体ネットワーク中のリアルワールドスイッチおよび他の要素に対する実施環境をインターフェイスするのに向けられる。このレベルで実行されかつすべての派生クラスによって受け継がれる内部状態の例はインサービス対アウトオブサービスおよび自由対使用中である。意図された派生クラスは電話 3 2 0（標準 2 5 0 0 セットに対する標準プロキシ）、音声応答ユニット（“VRUs”）3 2 2（音声応答ユニットに対する標準プロキシ）、IMT トランク接続 3 2 4（デジタルトランク（T 1 / E 1）回路に対する標準プロキシ）、およびリソース複合体 1 8 0 中の特殊な型のリソースに対応する、モデム接続 3 2 6（デジタルモデムに対する標準プロキシ）である。サービス制御構成要素が到来するサービス要求に役立つかも知れない好適な方法を、次に、図 1 0 をさらに参照して説明する。図 1 0 はサービス制御サーバー、例えば、汎用コンピュータ 4 4 0 の作動システム 4 3 5 内で実施している S L E E 用途 4 5 0、4 5 0' を有するサービス制御環境 4 3 0 の他の実施形態をとくに示している。

#### 【0075】

図 8 に示されるように、S L E E 4 5 0 はコール処理サービスおよび他の支持サービスを行うのに実行される少なくとも 5 つの型の論理プログラム（オブジェクト）を実施するように設計されたジャバ（商標）「バーチャルマシン」である。すなわち、1）特徴識別論理プログラム（“FD”）5 1 0。これは、まず切り換えプラットフォームからサービス要求を受信し、幾つかの利用可能な規準に基づいてコールを行うべきそのサービスを決定し、かつ次いでコールを処理するための他の適切なサービス論理プログラムを呼び出すサービス制御クラス／サービス識別クラス 2 9 6（図 7）の機能的なサブコンポーネントである。2）サービス論理プログラム（“SLF”）オブジェクト 5 2 0。これは、受信されたサー

ビス要求またはイベント用のサービス処理を行うサービス制御クラス252 (図7) の機能的サブコンポーネントである。3) ライン論理プログラム (“LLP”) 530。これは、ネットワークアクセスラインの現行の状態を維持するコール制御クラス250 (図7) の機能的なサブコンポーネントである。4) イベント論理プログラム (“ELP”) オブジェクト540。これは、すべての他の論理プログラムがそれにイベントを書き込むサービス制御/セッションマネージャークラス260 (図7) の機能的なサブコンポーネントである。5) コール論理プログラム (“CLP”) オブジェクト545。これは、コールの処理に伴われる他のすべての論理プログラム用の接続点を設けることによってコール全体の状態を維持するサービス制御/接続マネージャークラス302 (図7) の機能的サブコンポーネントである。これらの論理プログラムの各々は、説明されるように、一時的に例示されるかまたは継続することができる、好ましくは、ジャバ (商標) プログラミング言語に書かれたソフトウェア「オブジェクト」として具体化されている。IDNA/NGINサービス制御アーキテクチャは、これらのオブジェクトがMOCE/SCEに一度だけ書き込まれ、かつネットワーク中の何処でもあらゆる型のコンピュータおよびあらゆる型の作動装置についてSLEEに展開され得るように設計されている。

#### 【0076】

より大きな特徴により、FD510は、まず、例えば、スイッチがサービスがIDNA/NGINによって処理されることを識別するとき切り換えるリソース複合体からのサービス要求を受信し；2) サービス要求と関連付けられる情報を分析し；そして3) SLPがサービス要求を処理することができることを決定する静的サブコンポーネントである。好ましくは、FDは、コールされた番号、コールしている番号、開始スイッチID、開始しているトランクグループ、開始しているライン情報、およびネットワークコールIDを含むが、それらに限定されないリソース複合体から供給されたデータを受信するためのシステムタスクまたは例示されたオブジェクトであってもよい。NOSによって、FD510はコールを処理するために適切なSLP、CLPおよび開始しているLLPの例示を始める。好ましくは、FD510は、特定のコールまたはイベントに結合されてい

(39)

ない、継続するオブジェクトでありかつサービス制御 S L E E 4 5 0 において何時でも能動的に運転している。実施される分析の複雑さ、および F D に対する要求の量に依存して、負荷を分割しかつリアルタイム効率を保証するためにサービス制御 S L E E 4 5 0 中で能動的に運転している F D の 1 またはそれ以上の段階が存在してもよい。例えば、一方の F D は受信された S S 7 メッセージデータを分析するのに使用されることができ一方、他方の F D は A T M メッセージデータを分析するのに使用され得る。

#### 【 0 0 7 7 】

ライン論理プログラム (L L P) 5 3 0 は、1) ネットワークアクセスポイント、接続またはラインの現行の状態を維持し；2) 物理的なポイント、接続またはラインと連係する特徴に関してデータマネジメントに質問し；そして3) コール遮断、キャッチホン、コール送信、およびコール位置要求としてのオーバーフロールーチングのごときそれらの特徴を適用する機能的なサブコンポーネントである。コールを開始するラインと連係する L L P、以下で“L L P O” および点接続、またはコールが終了するラインと連係する L L P、以下で、“L L P T”がある。いったんライン論理プログラム段階 (インスタント) が例示されるならば、それ自体をスイッチ構造に記録する。説明されるように、ライン論理プログラム 5 3 0 はすべてのイベントデータをサービス処理の同一段階の E L P サブコンポーネントに送信する。

#### 【 0 0 7 8 】

動的サブコンポーネントはサービス処理の種々の段階に応じて動的に構成されかつサービス処理の段階が完了するとき破壊されるそれらのコンポーネントであり、そしてイベント論理プログラム (E L P) ；コール論理プログラム (C L P) ；およびサービス論理プログラム (S L P) を含んでいる。

#### 【 0 0 7 9 】

イベント論理プログラム (E L P) 5 4 0 はサービス処理の間中発生されかつサービスの実施の間中発生するすべてのイベントデータを記録するリアルタイムイベントデータを保持するのに使用される機能的サブコンポーネントである。好ましくは、イベント論理プログラムは、イベントが最初に受信されるときスイッ

(40)

チでのコール制御処理によって例示される。スイッチがサービス要求をN G I Nに送信するとき、その要求はイベントデータがそのコールに結合されたこの論理プログラムに送信され得るようにE L Pのアドレスに沿って通過する。イベント論理プログラムはサービス処理の同一段階中のすべてのサブコンポーネント、すなわち、コールに関連するC L P, L L P s, およびS L Pにアクセス可能である。各サービス処理コンポーネントがサービスの実行においてそのコールを処理するので、予め確立された規則にしたがって、N O Sによって、E L Pにイベントデータを書き込む。コールが完了されると、E L P中のイベントデータはデータ記憶装置またはログに書き込まれ、このデータ記憶装置またはログから、イベントデータが次いでビリング記録にコンパイルされかつビリング、通話量／使用報告、および他のバックオフィス機能用の下流のシステムに送られる。とくに、E L Pは、1) 特定のコールによって発生されたネットワークイベントを集め；2) イベントを適切なコールヒストリー記録、例えば、コール詳細記録（“C D R s”）、ビリングデータ記録（“B D R s”）、スイッチイベント記録等にフォーマット化し；3) 確認し、許可しかつ下流のシステム、例えば顧客請求書作成への将来の伝送のために、例えば、データマネジメントに情報を記憶する機能を実施する。理解されるべきことは、イベントがE L Pに書き込まれるのを決定する規則はサービス発生において確立される。イベントデータはフロードマネジメントおよびネットワークシステムによって追加的にアクセス可能である。

#### 【0080】

コール論理プログラム（C L P）545はサービス処理に伴われる各S L Pの状態を維持し、かつすべてのサービス（L P' s）の間のプロセスインターフェイスを設ける機能的なサブコンポーネントである。1実施形態において、C L Pはイベントサービス要求がコールに関してまず受信されるときF Dによって例示されるか、またはスイッチに置かれたコール制御コンポーネントによって例示され得る。代替的に、C L P 545はS L Pにプログラムされたトリガーポイントに応じて、サービス処理の間中幾つかの点においてS L P 510によって例示されることができ；この方法において、C L Pの例示はサービスに対して固有であってもよい。コール論理プログラムは例示の時間においてサービス処理の同一の

(41)

段階内ですべてのサブコンポーネント、すなわち、SLP a, LLP s およびELPのアドレスを受信する。CLPは、次いで、そのコールに関してSLP (s), LLP O, LLP T、およびELPを関連付けそしてサービス処理の同一の段階内でこれらのサブコンポーネントのすべてによってアクセス可能である。コールが完了されると、CLPは論理プログラムの分解プロセスを開始するコール完了のサービス処理の同一段階内でサブコンポーネントのすべてに通報する。

#### 【0081】

サービス論理プログラム(SLP) 520はサービスを実施するのに必要とされる論理を供給する動的サブコンポーネントである。SLPはコールよりむしろサービスに結合され、そしてコールに関する、サービス、およびそれに含まれる特徴を実行する。SLPがサービスに適用され得る特徴は、例えば、コールルーチンアルゴリズムおよびIVRサービスを含んでいる。SLPは頻繁に使用されるサービスの継続するオブジェクトにし得るか、またはFDによって要求されかつ例えば頻繁に使用されるサービスに関して、コール完了時中止されるとき例示されてもよい。一定のSLPが何時でも、或る時、または要求時のみ能動であるかどうかは、図11に示されるようなそのサービスに関するサービス管理によって発生される形状ファイル580によって決定される。好ましくは、SLPはサービス処理の同一段階内でCLPおよびELPにアクセスする。

#### 【0082】

すべてのSLPが特定のコールサービスに関連付けられずかつ幾つかのSLPは、他のSLPによって必要とされかつそれによって呼び出されるタスクに利用可能である。したがって、例えば、800サービス用のSLPはコールルーチン中継用のそのタスクを完了するためにライン情報データベース質問用SLPを引き出す必要があるかも知れない。また、SLPは他のSLPへのコール用コール処理の制御を通すことができる。好ましくは、サービス処理の単一段階の時には1つの制御SLPのみが実施しているべきである。SLPによって行われるサービスタスクの部分として発生されるあらゆるイベントデータは差蛇巣処理の同一の段階内でELPコンポーネント540に送られる。

#### 【0083】

S L Pは、これが実施すべき作動システムのすべての情報を含まないため作動システムにおいて直接実施されることはできない。そのうえ、S L Pがフォーマットおよび内容を変更することなく種々の作動システムにおいて実施される必要があるならば、S L Pと作動システムとの間のN O Sミドルウェアが作動システムを横切るS L Pの一致を維持するために設けられる。

#### 【0084】

図8にさらに示されるように、支持および作動機能に関するS L E E 4 5 0内で実施する他のプロセスは、S L E Eにおいて運転するサービスを負荷し、活性化し、消勢しかつ除去し、そしてさらに、そのS L E E内で運転する他のすべてのサービスを監視し、かつN O Sへ状態および利用データを報告する責任があるサービスマネージャー（“S M”）オブジェクト5 5 4；N O Sサービスとインターフェイスするのに使用されかつN O Sサービスを呼び出すのにS L E E内で運転するすべてのサービスによって使用されるN O Sクラスライブラリー、すなわち、N O SへのゲートウエーであるN O Sクライアントプロセス5 5 8；すべてのS L E Eリソースを結合することなく同時に実施するためにN G I Nサービスに必要とされる機能性を備えるスレッドマネージャー（“T M”）5 5 7；および図4（c）を参照して本書で説明されるようなD M 4 0 0の局部メモリ4 1 5およびメモリマネージャーコンポーネントとインターフェイスするのに使用されるデータマネジメントA P I（“D M A P I”）4 1 0を含んでいる。

#### 【0085】

図8に示されるようなS L E Eに負荷されたさらに他のサービス段階はサービスエージェント（“S a g”）段階5 5 9および本書においてさらに詳細に説明されるような、サービスノードにおいてサービス活動に利用される段階5 5 9と関係するスレッドマネージャー段階5 5 7を含んでいる。

#### 【0086】

図12（a）はS L E Eプロセスに主流入点を設ける（S L E E、ジャバ）プロセスステップを示している。図12（a）に示されるように、ステップ6 0 2は、D Mシステムが利用可能であり、N O Sクライアントプロセス5 5 8およびN O Sサービスとインターフェイスするのに使用されかつN O Sサービスを呼び

出すためにSLEE内で運転しているすべてのサービスによって使用されるNOSクラスライブラリーを設けるNOSマスタープロセスを含んでいるNOSサイトロケータ（位置決め）システムが論理的な名称およびオブジェクト参照登録を受信するのに利用可能であり、そしてサービス制御サーバー作動システム、例えば、ウィンドーズNT，UNIX，PC等が、例えば、メイン（）またはフォーク（）のごときブートストラップコールを認識することによりSLEEプロセスを開始することかできると考えられる。理解されるべきことは、NOSマスターコンポーネント560（図8）はコンピュータの作動システム、NOSクライアントプロセス556および他のシステムコンポーネント571と直接インターフェイスするということである。好ましくは、各SLEE上でNOSクライアントオブジェクト558とインターフェイスしかつNOSサービスを供給するためのすべてのNOSクラスライブラリーを含んでいるネットワークまたは局部ノード上に置かれたNOSマスタープロセス560が存在している。次に、ステップ604において、サービス制御形状ファイルは形状オブジェクトを形成するようなファイルを記述しそしてステップ606で示されるようなキーバリュー対を含んでいるハッシュテーブルを含むことも可能である。SLEEは2つのパラメータ、すなわち、名称および形状ファイルを受容する。名称パラメータはSLEEのこの段階を識別するためのNOSロケータサービスによって使用される、すなわち、NGINロケータサービス（ステップ612）によりそれ自体を登録するためにSLEEによって使用される独特のNGIN名称ストリングであり、そして形状ファイルはそのサイトロケータを見つけるためにロケータサービスによって使用される。例えば、このテーブルはSLEE形状特性を見いだすのに使用され得る。NOSがCORBAを実行するとき、ベースCORBA機能性は次にステップ608において初期化される。次に、ステップ610において、SLEEクラスローダークラスが例示されかつNOSロケータプロキシサービスがステップ612で示されるようにSLEE内で例示される。次に、ステップ615において示されるように、サービスマネージャー（SM）クラスがクラスローダークラスを経由して負荷され、例示されかつ局部NOSと結合される、すなわち、SMオブジェクトは局部プロキシNOSロケータサービスオブジェクトにより登録さ

(44)

れる。理解されるべきことは、局部ロケータサービスはNGIN領域において他のロケータサービスへのサービスマネージャー登録に伝搬するということである。図12(b)を参照して説明されるように、サービスマネージャーオブジェクトがロケータサービスにより登録された後、SLEEへの／それからのサービスを負荷し、付勢し、消勢しかつ除去することができる。最後に、ステップ618において示されるように、プロセスイベントループが実施され、このループはSLEE運転を保持しかつNOSイベントが本書でより詳細に説明されるようにサービスマネージャー(“SM”)およびサービスエージェント(“Sag”)オブジェクトを通して到来するようにSLEEにNOSイベントを処理させ得るSLEEスレッドである。

【0087】

図12(b)は、図12(a)、ステップ615を参照して上記で議論されたように例示されるサービスマネージャーオブジェクト段階554によって実施される(サービスマネージャーImpl. ジャバ) プロセスステップを示している。好ましくは、SMオブジェクトはNOSのためにサービスマネジメント演算を行うためのORBインターフェイスを実行する。プロセスはSLEE内のサービス、例えば、(負荷)、(運転)、(スタート)および(停止)方法を負荷し、付勢し、消勢し、運転しかつ終了するようにSM段階によって取られるステップを示している。NOSによってSMオブジェクト段階に通されたパラメータは所望されるサービスの論理基準およびNOSがNGIN局部リソースマネージャー(LRM) サイトロケータによりサービスを登録すべきかどうかまたはサービスがNOSによりそれ自体登録する責任があるかどうかを示しているブールフラッグを含んでいる。ステップ620に示されるように、サービスを負荷する要求がまず受信され、そしてステップ622でプロキシネーミングサービスが処理される。次いで、ステップ624で、登録されたサービス、例えば、1~800コレクト(“18C”)がすでに負荷されているかどうか、すなわち要求されたサービスを具体化しているオブジェクトが例示されているかどうかについて判断がなされる。要求されたサービスに関するオブジェクトがすでに例示されているならば、その場合に、NOSはステップ626で物理的オブジェクト段階を配置する



(45)

ためにそのサービスのオブジェクト基準に戻りそしてプロセスはステップ632に戻る。要求されたサービス用のサービスオブジェクト、例えば、18Cがすでに例示されていないならば、その場合に、クラスローダークラスは、他のSLPsおよびSIBBsを含んでいる、要求されたサービスが依存するすべてのクラスを負荷するために帰納的負荷を実行するステップ625において例示される。帰納的負荷は、例えば、局部メモリからの局部形状ファイルに言及することにより可能である。とくに、クラスローダーがJVMにこれらのすべての依存するクラスにおいて帰納的に負荷するかどうかを示すフラグが通される。第1の段階においてサービス用のクラスを負荷するとき、理解されることは、全般的なサービスエージェントクラスがそれがまだ負荷されていないならば負荷され得るということである。次いで、ステップ625でクラスをすべて負荷した後、ブールレジスタフラグが、サービスがそれ自体局部NOSネーミングサービス（プロキシ）により登録しなければならないかどうかを判断するためにステップ625でチェックされる。ブールレジスタフラグが、例えば、正確に、設定されたならば、その場合に、サービスは、ステップ630で示されるように、NOSネーミングサービスを登録するような責任を有している。そうでなければ、プロセスはSAgクラスが例示されているステップ632に継続し、そしてサービスエージェントオブジェクト段階558（図11）と特定のサービスの間で、すなわち、SLPオブジェクトにおいてサービスエージェント段階に通すことにより、関係がなされる。次いで、ステップ635において、新たなSLEEスレッドが、記載されるような方法において、作られそしてSLEEスレッドがサービスエージェントを運転するように引き起こされる、すなわちSLEEスレッドをサービスエージェントと関係する。最後に、SMプロセスが励起されかつプロセスはSLEE、ジャバプロセスに戻る。SMに設けられた方法を経由して、それはそのSLEE内で運転している他のすべてのサービスを監視し、かつNOSへ状況および利用データを報告する責任がある。

【0088】

SMプロセスに関して、（SLEEクラスローダー、ジャバ）の引用が図12（c）を考慮してより詳細に説明される。とくに、SLEEクラスローダークラ

スはJVMのクラスローダークラスの特権化されたクラスでありかつそれを拡張している。それはネットワークにわたって負荷されるようなクラスを許容することによりシステムクラスローダーの行動を拡張している。したがって、図12(c)の第1のステップ686として、クラスローダーが、まず、クラスがすでに負荷されかつ定義されている場合に見るようにSLEEの段階と関係されたその局部メモリをチェックする。クラスがすでに負荷されたならば、その場合に、プロセスは戻る。クラスが負荷されていないならば、その場合に、ステップ688において、メッセージがNOSを経由してクラスが負荷するのに利用可能であるならば局部データ記憶(DM)をチェックするために送られる。例えば、SLEEクラスローダーはJDBCデータベース接続性を使用して関連のデータベースからクラスを検索することができるが、しかしながら、理解されることは、JDBC APIを支持するあらゆる関連のデータベースからクラスを検索することができるということである。サービスクラスが局部データ記憶に見いだされないならば、その場合に、SLEEクラスローダーはステップ689において局部ファイルシステムをチェックする。クラスがデータ記憶、または局部ファイルシステムに見いだされるならば、クラスは、ステップ690において示されるように、フェッチされる。次いで、ステップ694において、定義されたクラス方法がJVM実施環境に利用可能であるそのクラスを作るように引き起こされる。とくに、(定義されたクラス)方法はそのサービスを行うために特定されたクラスの各々を帰納的に通り抜けかつクラスClassの段階にバイトのアレイを変換する。この新たに定義されたクラスは次いでクラスClassにおいて新たな段階方法を使用して作られ得る。この機能性はSLEEを負荷させかつ新たなサービスを例示しかつまだ全般的に残っている。好ましくは、ステップ695に示されるように、方法はクラスが負荷される次回にメモリヒットがあるように局部メモリをポピュレートするように求められる。

#### 【0089】

好適な実施形態において、これらの例示されたオブジェクトの各々は、一般に以下のストリング、すなわち、

．．． サイトレベル、SLEE番号、SLP名称．．．

によって一般的に例示される、ネーミング取り決めにしたがって、NOSロケータサービス、すなわち、LRM577によりそれら自体登録する。この場合に、サイトレベルはNGINサービス制御サーバー440の物理的位置に関する情報；SLEE番号はそのオブジェクトが例示された特定のSLEE、例えば、SLEE#1；およびSLP名称はサービスの局部的名称、例えば、特徴弁別器#1である。ストリングは値同様に、「バージョン番号」を含むことができる。登録名称はNGIN領域において他のロケータサイトに伝搬され；これによりそれは登録プロセスおよびNOSリソースマネジメント機能性（記載されるような）でありそれによりNOSコンポーネントが知られそれらのプロセスは、それらが展開される場合に、かつサービスが容易に利用可能である場合に展開される。

#### 【0090】

クラスローダーによって作られたオブジェクトの方法およびコンストラクターは他のクラスを参照することができる。参照されるクラスを決定するために、ジャババーチャルマシンがクラスを最初に作ったクラスローダーのロードクラス方法呼び出す。ジャババーチャルマシンがクラスが存在するかどうかおよびそれがそのスーパークラスを知るために存在するかどうかを判断するためにのみ必要であるならば、「決定（リゾルブ）」フラッグが不完全に設定される。しかしながら、クラスの段階が作られているかまたはその方法のいずれかが作られているならば、クラスは、また、決定されねばならない。この場合に、決定フラッグは正確に設定され、そして決定クラス方法が呼び出される。この機能性はサービスによって参照されるクラス／SIBBs／ジャバビーンズがまたSLEEクラスローダーによって決定されることを保証する。

#### 【0091】

図12（d）は例示時のサービスエージェントクラスプロセス流れを示している。ステップ639に示されるように、第1ステップはサービスエージェントと関係しかつ図11にTMオブジェクト段階557として描かれているスレッドマネージャー（“TM”）オブジェクトの例示を含んでいる。記載されたように、スレッドマネージャーオブジェクトはサービス要求毎に新たなSLEEスレッドを作るように機能しているスレッドファクトリーのごとく行動するように例示さ

れ得る（スレッドマネージャー）クラスに、または高いスレッド発生潜在能力を備えたマシンで運転するとき所望されるスレッドウェアハウスに基礎を置いている。次いで、ステップ640において、サービスと関係するSAがその（運転）クラス方法を経由してプロセスイベントループに入り、そしてサービスと関係するコールイベントを受信するのに今や備えている。

#### 【0092】

図12（e）を参照すると、その（開始）、（継続）および（終了）クラス方法を経由してNGINサービスにゲートウエーを設けるサービスエージェントの詳細が示されている。SLEE内の各々のサービスはサービス段階（コール段階）を管理しかつサービス段階にイベントを急送する責任があるクラスに基礎を置いている関連のサービスエージェントオブジェクトを有している。図12（e）に示されるように、SAgオブジェクトがサービスマネージャー（負荷）方法によって例示されかつ運転している後、SAgの（開始）方法はサービスが受信されることを要求している新たなコール毎に引き起こされる。とくに、図12（e）に示されるように、ステップ641において、ティッド（tid）、オリッド（orid）コール識別子パラメータおよび例えば、ネクストジェネレーションスイッチ（“NGS”）として本書で言及されるIDNA/NGINからのイニシャルアドレスメッセージ（“IAM”）によって設けられるような、そのコール用のサービス処理に関連付けられるイベント情報を含んでいるメッセージの流れが最初にSAg開始方法に通される。次いで、ステップ643において、メッセージ流れは、例えば、そのサービス段階に関連付けられる重要な情報を抜き出すように（デコード）方法を引き起こすことにより復号される。加えて、コール背景データを管理するのに使用されるコール背景オブジェクト段階が抜き出されたメッセージ情報を受信するように作られる。ステップ645に示されるように、開始方法において、新たなスレッドが、図12（g）を参照して本書で記載されるように、スレッドマネージャー段階の割り当て方法を引き起こすことによりそのコールに割り当てられるか、または、スレッドはその差一美委用の幾つかのスレッドが時間を超えて例示されるならばスレッドのプールから引き出される。他の方法で、SAg（連続）方法が引き起こされるならば、そのコール用に割り

当てられたスレッドに対応するオブジェクト基準は戻される。

【0093】

よりとくに、スレッドマネージャオブジェクトは、好ましくは、セッション *i d s* に基礎を置いたスレッドを管理するスレッドマネージャに基礎が置かれている。2つの方法、（割り当て）および（解放）が、それぞれ、スレッドを割り当ておよび解放するために設けられる。割り当ておよび解放の両方はスレッド識別に使用され得るキーとして独特の識別子を期待している。独特の識別子はコールを受信したNGSスイッチによって設定される処理ID（“*T i d*”）、およびコールオリジネーターを識別するオブジェクト基準ID（“*O r i d*”）を含んでおりかつコール段階を識別するのに使用される。図12（f）はスレッドマネージャクラスの（割り当て）方法の作動の詳細を示している。図12（f）に示されるように、ステップ660において、コール処理を独特に識別するための*T i d*および*O r i d*識別子はプロセスに通されかつ独特のキーが識別子に基礎を置いて発生される。次いで、ステップ662において、質問が、例えば、キーバリュース対のハッシュテーブル（寄せ集め表）をチェックすることにより、キーがすでに存在しているスレッドを識別するかどうかについてなされる。キーがサービススレッドがコールに対してすでに割り当てられたことを意味して認められるならば、その場合に、ステップ664において、スレッドマネージャはハッシュテーブルを考慮に入れた後SLEEスレッド段階（スレッドオブジェクト）に復帰する。他の方法で、ステップ663において、例示されたサービススレッドの数に一致するカウンタが増分され、そしてシステム負荷を監視するような努力において、ステップ665で、そのサービスのスレッド段階の最大値が超過したかどうかについて決定がなされる。そのサービス用のスレッド段階の最大値が、例えば、サービス形状ファイルに見いだされる最大サービス段階とのカウンタ値との比較時、超過されたならば、その場合に、ステップ667において、メッセージが、例えば、同一のサイトで実施するか、または他のサービスノード位置で例示された他のSLEEにおいて利用し得るサービス用の他の段階を求めることができるようにNOSに放出され、そしてプロセスは戻る。SLEEスレッド例示プロセスに関して、図12（g）を参照して本書で記載されるように、

その優先イベントキューの初期化である。そのサービス用のスレッド段階の最大値が超過されないならば、その場合に、ステップ 668 において、そのサービス用のスレッド段階のしきい値が超過したかどうかについて判断がなされる。そのサービス用のスレッド段階のしきい値が超過されたならば、その場合に、警告がサービスしきい値が達成された NOS 局部リソースマネジメント機能に発せられる。最後に、ステップ 670 において、ステップ 668 での出力に関係なく、要求されたサービス用の新たな SLEE スレッドが割り当てられ、優先イベントキューがその要求されたサービスに関して例示されそしてスレッドがそのサービス用の SAg 段階に戻されている制御により開始される。

#### 【0094】

図 12 (e) に示されたようなサービスエージェント (開始) 方法機能性に戻って、スレッドマネージャーがサービス段階用のスレッドを割り当てた後、スレッドに関連するオブジェクト変数がステップ 646 で初期化され、そして要求されたサービスの新たなオブジェクト段階が (クローン) 方法を引き起こすことにより例示される。次いで、ステップ 648 において、新たなクローン化された SLP 段階が新たに割り当てられたスレッドに設定される。次いで、ステップ 650 において、そのコール段階と連係することが必要とされるイベント情報、例えば、入力メッセージ流れから抜き出されたすべての IAM 情報が存在するかどうかについて判断がなされる。新たにクローン化された SLP と連係されるイベント情報があるならば、その場合に、ステップ 652 で示されるようにスレッド上に押し出される。スレッド上に押し出されるようなイベント情報が存在するかどうかはともかく、その SLP に新たに割り当てられたスレッドが開始され、SA (連続) 方法によって処理されるサービス関連のイベント情報の非同期到達を待っている。述べられたように、そのコールに割り当てられた SLEE スレッドは処理の間中受信されたすべてのサービス関連のイベント情報を保持するために優先イベントキューを維持している。サービス処理に関連のすべてのイベントは関連の優先権を有しかつスレッドはその優先権、すなわち、そのサービスイベントキューにおけるその位置にしたがってイベント情報の処理を管理する。最後に、ステップ 654 において、スレッドイベントループがそのコール段階に関して開

(51)

始される。

【0095】

理解されるべきことは、SA（連続）方法が実質上図12（e）に示された（開始）方法と同一であるということであり、その差異は、図12（e）を参照して上記で議論されたように、そのコール段階に関してすでに例示されたサービスプロセススレッドとのリアルタイムサービス関連のイベントを運ぶのに向けられる。したがって、サービスエージェント連続方法はコール段階のイベントおよび識別パラメータを受信し、受信されたイベント用のTid、Oridパラメータと関係されたサービススレッドを再び割り当て、そしてスレッドのイベント優先キューへイベントを押し出す。理解されるべきことは、SAgおよびSMクラスの両方がNOSへのIDLインターフェイスからなっているということである。サービス（SLs）はこのようなインターフェイスを持たないが、しかしながら、そのSAgインターフェイスを会してシステムと広く連通することができる。リアルタイムサービス処理の間中、SLEE450は以下を、すなわち1）サービス処理の間中SLPおよびSIBBでの指示を解釈し；2）SLPの指定された段階へ到来するイベントを供給し；3）追跡フラッグが設定されるならば、追跡データを発生し；4）SLP、SIBB、およびSLEEレベルでオンされた追跡を許容しかつ追跡データを特定された出力に送り；5）SLEE使用データを発生しかつ運転時間使用データを特定された出力に送り；6）遠隔通信マネジメントネットワーク（TMN）インターフェイス用の例外的なデータを発生し；7）TMNインターフェイス用のパフォーマンスデータを発生し；8）SLPまたは実用プログラムの新たな段階を加えるためのメッセージ／要求を受信しかつこのような新たなSLPまたは実用プログラムをサービス処理を中断および劣化することなく加え；そして9）負荷分割用の多重サービス制御段階によって同一のサービスを支持することを実行することができる。

【0096】

サービス段階が処理を終了したとき、サービスの終了、またはサービスと連通して他のプロセスを開始する。いずれの場合においても、SAg（終了）方法が呼び出され、これはそのコールと関係したスレッド段階を終了するように機能す

る。これは、スレッドマネージャー（解放）方法を引き起こし、コール段階を独特に識別する `Tid` および `Orid` 識別子に通し、イベントをスレッドイベントキュー上に押し出し、そしてコールを解放し、すなわちスレッド段階を終了しおよび／またはスレッド段階をスレッドプールに配置することによりなし遂げられる。

#### 【0097】

好ましくは、`SLEE` スレッドクラス段階はすべての `SLEE` リソースを結合することなく同時に実施するような `IDNA/NGIN` に必要とされる機能性を備え、そして協働リソース分割を容易にする。とくに、サービスの1段階と `SLEE` スレッドの1段階を連係する `SLEE` により `SLEE` スレッドとサービス段階との間に1対1のマッピングが存在する、すなわち、サービスによって取り扱われる各コールに関して、コールと連係する `SLEE` スレッドの1つの段階がある。`SLEE` スレッドは、また、処理 `id (tid)`、オブジェクト基準 `id (orid)`、オブジェクト基準、例えば対等者およびエージェントの両方、`SLP`、および `SLP` と連係する優先イベントキューを収納することによりサービス用のデータウェアハウスのように作用する。とくに、`SLEE` スレッドは、2つのキーインターフェイス、すなわち、サービスエージェントが `SLEE` スレッド上にイベントを押し出すのを可能にするためのプッシュコンシューマー；およびサービスがそれらの関連のスレッドからイベントを引っ張ることを可能にするプルサプライヤーを実行することによってサービス（`SLP`）とサービスエージェントとの間のイベントチャンネルのように作用する。記載されるように、各 `SLEE` スレッドは、記載された方法において、`NGIN` イベントを待ち行列に入れるための優先イベントキューの段階を有している。

#### 【0098】

好ましくは、（優先イベントキュー）クラスはサービス（`SLP`）と連係するイベント（`NGIN` イベントの派生クラス）を待ち行列に入れるプラットフォーム独立クラスである。図12（f）の、ステップ667、670を参照して示されたように、各 `SLEE` スレッドオブジェクトはイベントのハッシュテーブルからなることも可能である優先イベントキューの段階を例示している。イベントは下



(53)

降順位において待ち行列に入れられることができ、イベント優先性はNG IN イベントにおいて定義されておりかついずれにしても10から1の範囲にあり、10は、例えば、最高の優先である。したがって、各スレッドは処理に利用可能／利用し得ないイベントの数を追跡することかでき、かくして完全なサービス処理一致を可能にしている。

#### 【0099】

図12 (g) は、ステップ675に示されるように、スレッドによって受信されているイベントの優先性、および優先イベントキューへのイベントの供給を確認するために論理をカプセル化する（ポストイベント）方法を示している。図12 (g) に示されるように、これは、ステップ678で処理されるような優先キュー上で次のキューの優先性と押し出されたイベントの優先性を比較し、押し出されたイベントがステップ680で処理（もしあるならば）されるようなキューにおける次のイベントの優先性より大きい場合に判断し、そしてステップ682aで示されるように処理されるべき次のイベントとしてそれを設定するようにキューの頂部に押し出されたイベントを配置するか、または、キューを通してループにしかつステップ682bで示されるように、イベントがその優先性にしたがって記憶されるべきであるキューの位置を決定することによって実質上なし遂げられる。次いで、ステップ684において、SLEEスレッドは、システムから処理時間が割り当てられるとき最高の優先性の次のイベントを処理する。

#### 【0100】

より詳細には、プルサプライヤーインターフェイスは、イベントデータが利用し得るかまたは例外が生起されるまで遮断しかつイベントデータを顧客に戻す「プル」作業、または、遮断しない「トライプル」作業を引き起こすことによりサプライヤーからデータを要求するような顧客のための作業を支持するためにSLEEスレッドによって実行される。すなわち、イベントデータが利用可能であるならば、イベントデータを戻しかつハズイベント（hasEvent）パラメータを正確に設定し；イベントデータが利用不能であるならば、ハズイベントパラメータを不完全に設定しかつゼロに等しい値が戻される。したがって、SLEEスレッドはイベントサプライヤーとして作用しかつサービス（SLP）は顧客の

役割を引き受ける。サービス (S L P) は S L E E スレッドからイベントデータをフェッチするために S L E E スレッドプルまたはトライプルを使用する。サービスはイベントデータなしに継続できないならばプル作業を使用し、そうでなければ、トライプル作業を使用する。

#### 【0101】

プッシュコンシューマーインターフェイスは S L E E スレッドによって実行されかつスレッドへのプッシュ作業を引き起こしかつイベントデータをそのスレッド優先イベントキューにパラメータとして通すことにより顧客へイベントデータを通信するためのサプライヤー用作業を支持する全般的なプッシュコンシューマーインターフェイスである。したがって、S L E E スレッドはイベント顧客として作用しかつサービスエージェントはサプライヤーの役割を引き受けている。サービスエージェントはイベントデータを S L E E スレッドへ通信するために S L E E スレッドプッシュ作業を使用している。「キル (k i l l)」サービスイベントは最高の優先性からなることができる。イベントの優先性は怠われてもよいが、または、新たに作られたイベントクラスが設計されるとき、サービス作成において確立されてもよい。

#### 【0102】

記載されたように、特定のサービス用のサービスエージェント段階は、そのコールのために作られたサービススレッド段階への／それからのサービス処理の過程中に受信されかつ発生されたすべてのイベント伝える。例えば、ノードにおいてスイッチによって発生された最初のイベントは (サービスリクエストイベント) からなることができ、このクラスは I D N A / N G I N サービス制御へ最初のサービス要求をかつとくに、サービス要求が始められる時間；要求がそれから開始されるスイッチ I D；コールが引き起こされるポート I D；コールが引き起こされる端末設備 I D；呼び出している者の番号；呼び出された者の番号等のごとき関連の初期コール背景情報を運ぶ責任がある。N G I N イベントを拡張している (コネクトイベント) サブクラスは接続が発生する時間；呼び出している番号が接続されるステーション番号を報告し；および A T M - V N E T サービスの背景において、到来すバーチャルパス I D および出て行くバーチャルパス I D s を

報告することができる。NGINイベントを拡張している（解放イベント）サブクラスは解放イベントを報告することが可能である。例えば、ATM-VNETサービスの背景において、解放は、呼び出しているまたは呼び出された者がコールを終了するとき、または使用者クレジットが切れているとき発生され得る。このようなクラスは、解放イベントが発生される時間；解放イベントが発生する原因および呼び出しているおよび呼び出された者から解放イベントが発生される時間への所要時間を決定するためのSIBBSを実行することができる。これに関して、NGINイベントを拡張している（終了イベント）サブクラスがNGINからNGSへ終了メッセージを帆走するのに使用されてもよい。このメッセージを受信時、スイッチは分解接続プロセスを開始し得る。（モニタ解放イベント）サブクラスはNGINイベントを拡張しかつ解放指示の受信時NGINに解放指示を送るようにNGSに向いているNGSへのメッセージを送るのに使用される。NGSがモニタ解放メッセージを受信するとき、（ユニノーティファイイベント）サブクラスがオリジネーター（呼び出し者）への通知を送って引き起こされ得る。（モニタ接続イベント）サブクラスはNGINイベントを拡張しそして接続メッセージを受信されるときNGINへイベントを送るためにNGINからNGSへメッセージを送るのに使用されるサブクラスである。

#### 【0103】

記載されたように、リアルタイムサービス処理の文脈において、データマネジメントデータ検索および更新機能はサービス処理の間中DMによって記憶されたデータにアクセスするような能力を包含している。

#### 【0104】

好適な実施形態において、ある特定のサービス・ノードで、DMは、サービス処理中に、たとえばNOSを介して、SLEE中の実行被管理オブジェクト・インスタンスからのデータ要求を受信する。データ管理（Data Management）は、そのデータ要求を理解できない場合、要求元（たとえば被管理オブジェクト）に特に通知する。このデータ要求がデータ・エンティティの検索に対するものである場合、データ管理は、要求されたデータを要求元に（たとえばNOSを介して）返す。なお、単一のリポジトリ中、または複数のリポジトリにまたがってデー

タを操作および問い合わせるために必要なサポートはDMによって供給される。さらに、データ管理は、複数のリポジトリにまたがる問合せの結果の収集または対照をサポートする。DMはデータ検索要求中にある要求されたエンティティの名前の場所を特定できない場合、DMはNOSコンポーネントに通知する。また、NOSコンポーネントは、データ・エンティティの検索中にデータベース障害が発生した場合に通知される。さらに、データ管理は、要求元（サービス制御オブジェクトを実行する）に、有効な名前から特定のデータ・エンティティを検索できない旨を通知する。データ要求がデータ・エンティティの更新に対するものである場合、データ管理は、データ・エンティティを更新して、複製が要求されるか否かを決定する。DMは、データ要求中で指定されたデータ・エンティティを更新することができない場合、要求元に通知し、さらに、データ更新要求中における要求されたエンティティの名前の場所を特定することができない場合、NOSに通知する。NGIN動作中において、DMは、NOSに、データ・エンティティの更新時のデータベース障害について通知する。データ要求がデータ・エンティティの削除に対するものであれば、DMはそのデータ項目を削除し、他のリポジトリ上でトランザクションの初期化が必要か否かを決定する。

#### 【0105】

図4（c）は、全体として、データ管理コンポーネント400の機能アーキテクチャを図示している。データ管理コンポーネント400は、リアルタイム呼処理に対して呼び出しサービス・データがサービス・ノードで使用可能となるようにするサービス制御サーバ・コンポーネント405と；個別データベース・サーバとして具現化されSAによって保持されるデータの被選択サブセットを格納および分配するデータベース・コンポーネント407とを備える。特に、サービス制御サーバ・コンポーネント405は、実際のデータ管理アプリケーションであるデータ管理（DM）クライアント409と；DMアプリケーションとリンクされDMアプリケーションがSAからデータを獲得するために使用するインタフェースであるDM API 410と；ローカル・キャッシュ・ストラテジにしたがって、呼処理に対して使用可能なDBOR抽出からのデータの一部または全てを格納するために使用されるサービス制御サーバ上の共有メモリであるローカル・

キャッシュ415と、ローカル・キャッシュ・ストラテジをインプリメントすることによってローカル・キャッシュの状態を維持し、DMサーバと通信を行ってDBOR抽出からデータを検索するキャッシュ・マネージャ420とを備える。データベース・コンポーネント407は、当該ノードでのサービス実行時に被管理オブジェクト・インスタンスによって使用されるデータを有する1つ以上のデータベースを含むDBOR抽出427と；SAが保持する情報の被選択サブセットを管理するDBOR抽出マネジャー426と；サービス・アドミニストレーションから受け取ったデータをDBOR抽出マネジャー426へ入力するSAクライアント422と；SAクライアント422とSAのデータ分散プロセスとの間のプロセス・インタフェースであるDDAPI424と；DBOR抽出マネジャー426からのデータ抽出を全般的に取り扱うデータ管理サーバ425とを備える。

#### 【0106】

ここで、データ管理動作について、図4(c)および図8に関してさらに詳細に説明する。SLEE内において、数種類の機能は、被管理オブジェクト(SIBB、SLPなど)およびNOSに限らず、データ管理400からのデータを必要とする場合がある。そのそれぞれを、サービス制御SLEE中で実行するDMクライアントとして図4(c)に図示する。DM API412はデータ管理とのインタフェース接続のために全DMクライアントに対して共通メッセージ・セットを供給するため、DMクライアント410は、DM API412を使用してデータに対する要求を行う。また、DM API412はデータが必要とされる特定の場所をDMクライアントからカプセル化するが、これは、このデータがローカル・キャッシュ415中またはDBOR抽出427中のみに格納される場合があるからである。DMクライアント410は、論理名によってデータを要求し、DM API412は、そのデータがローカル・キャッシュから検索可能か、またはDBOR抽出からDMサーバを介してデータを要求する必要があるかを決定する。ローカル・キャッシュ415は、制御サーバ405中に供給される各SLEE上に動作する各プロセスに対して使用可能となっている共有キャッシュであること、すなわち、たとえば1-800プロセス・キャッシュ、ルーティン

グ・マネジャー・キャッシュなどの異なる適用に対して1つ以上のローカル・キャッシュが備えられており、各供給キャッシュがそれ自体の各キャッシュ・マネジャーを有するkとが好ましい。

#### 【0107】

DMクライアント410がデータを要求すると、まず、DM APIはローカル・キャッシュ415を調査して、要求されたデータがそこに格納されているか否かを確認する。要求されたデータがローカル・キャッシュ415中に格納されている場合、DM APIはその要求されたデータを検索して、ハッシング・キーおよびアルゴリズムまたは索引順次アクセス方式など、標準データ検索技術を使用して、それをDMクライアント410に供給する。

#### 【0108】

要求されたデータがローカル・キャッシュ415に格納されていない場合、関連キャッシュ・マネジャー420は、DBOR抽出427からDMサーバ425を介して、データを検索する。特に、DM API412は、キャッシュ・マネジャー420に、特定のデータを必要とすることを通知し、キャッシュ・マネジャーは、要求をDMサーバ425に送信することによって応答する。その後、DMサーバ425は、データベース・アクセスのためにDBOR抽出マネジャー426を使用して、要求されたデータをDBOR抽出から検索する。DMサーバ425は、要求されたデータをキャッシュ・マネジャー420に返送し、キャッシュ・マネジャーはデータをDM API412を介してDMクライアント610に供給する。また、キャッシュ・マネジャーは、要求されたデータをローカル・キャッシュ415に書き込むが、これはサービス要求とそれらが動作しているコンピュータの機能、つまりメモリ容量との両方に依存するローカル・キャッシュ・ストラテジに依存する。これらの使用は、サービス・アドミニストレーションによって生成されるサービス・プロファイルおよびコンピュータ・プロファイルから獲得される。IDNA/NGINのDM400のためのデータキャッシュ・マネジャー・コンポーネントは、「クライアント側キャッシュ」ストラテジを、各サービス・ノードで 사용할ことが好ましい。

#### 【0109】

(59)

では、IDNA/NGINネットワーク・オペレーティング・システム（NOS）コンポーネント700について、図8～10を参照して、より詳細に説明する。上述のように、NOS機能は、ISNA/NGINシステム170に対するプロセス間通信機能と、オブジェクト・コネクティビティ機能と、ローカルおよびネットワーク全体リソース管理機能とを備える。全てのNGINプロセスは、広範囲に分散するアーキテクチャ中のさまざまなハードウェアおよびオペレーティング・システム・プラットフォーム上で実行するため、NOSは全プロセス間でプラットフォーム独立型通信および場所独立型通信を実現する。特に、NOSはいくつかの機能サブ・コンポーネントを備え、サービス制御、サービス・アドミニストレーション、およびデータ管理との間のインタフェースを含む、全NGINプロセス間のインタフェースを提供する。また、NOSは、スイッチ呼制御とサービス制御との間のインターフェースであり（図5）、同一SLEE上で動作する2つ以上のプロセスが互いに通信可能なようにする。

#### 【0110】

図8～10に示すように、NOS機能サブ・コンポーネントは、1) データ・オブジェクトおよびサービス・オブジェクトに対する論理名を、要求されたオブジェクトが動作しているコンピュータ（ネットワーク・アドレスとして）とメモリ・アドレスとの両方を識別する物理的アドレスに分解する名前変換（NT）プロセス570と；2) SLEEとサービス・ノードとで実行しているリソース・ステータスを追跡および維持するローカル・リソース管理（「LRM」）プロセス575、577と；3) NGINネットワーク全体における全サービス・ノード・リソースのステータスを維持し、プロセス間通信を提供するグローバル・ネットワーク・リソース・ステータス（「NRS」）プロセス590と；4) 共通オブジェクト要求ブローカー・アーキテクチャ（CORBA）対応ORBによって提供されるものなどのオブジェクト・コネクティビティを提供して異なるコンピュータ・プラットフォーム、APIメッセージ・セットおよびインターネット・プロトコル（IP）通信にまたがるオブジェクト間の通信を特定のリアルタイム呼処理動作上件を満足する以上の方法で、可能とする1組のサービスとを含む。たとえば、典型的な1-800番号「コレクト・コール」イベントを処理する

典型的な応答時間は、約50～100msecでなければならない。

#### 【0111】

ここで説明するように、NOSコンポーネント700は、たとえばOrbix（登録商標）が提供しMA、ケンブリッジ、およびアイルランド、ダブリンのIONAが開発したオブジェクト・コネクティビティに対して、CORBA対応ORBを使用してインプリメントすることが可能である。ORBは、物理的地址への論理名の対応付けを可能とする名前サービスを提供することによって、異なるコンピュータ・プラットフォームにまたがるオブジェクト間の通信を提供する。

#### 【0112】

システム・ブート時、SLEE450は起動され、その環境内において、NOSクライアント・コンポーネント558およびサービス・マネジャー・プロセス・コンポーネント554とのインスタンスを起動する。SM SLP554は、即時にインスタンス化されるサービスの論理名を含むコンフィギュレーション・ファイル580から、他のコンポーネントに対する論理名を検索する。その後、その論理名をORB名前サービスに供給し、ORB名前サービスは、その論理名を物理アドレスに対応づける。ORBはこの時点からサービス・オブジェクト・コネクティビティを保持する。また、ORBサービスは、他のサービスの登録に対しても使用される。SLEE上で開始された各サービスは、自分自身をNOSに登録し、それらの登録によって、ORBは論理名に対する物理アドレスを識別する。

#### 【0113】

対話型オブジェクト間でのプラットフォーム独立型通信をインプリメントするために、インタフェースが定義されるが、これはインタフェース定義言語（「IDL」）によって可能である。CORBAはIDLを現在サポートしているが、リアルタイム処理に対する動作上件が満足される限り、リモート・メソッド呼び出し（RMI）プロトコルなどの他のオブジェクト指向通信技術がインプリメントされることも可能である。特に、NGINコンポーネントのそれぞれに対するインタフェースがその作成時に定義され、それらを、図9に図示されるような



(61)

、ローカルLRM575と関連づけられた永続型データ・ストアまたはライブラリ（不図示）中に格納することによって、動作時に使用可能となる。サービスはこのライブラリに問合せを行うことが可能とされ、新規のオブジェクト・インタフェースについて学習する。NOSクライアント・プロセス558およびNOSマスタ560（図8）は、NOSサービスとのインタフェース接続のために使用され以下で詳述されるNOS NTおよびLRMサービスを呼び出すために当該SLEE内で動作する全サービスによって使用されるNOSクラス・ライブラリである。

#### 【0114】

ここで図9を参照すると、1つ以上のSLEE450および450'を実行するコンピュータ上に常駐するNOS機能サブ・コンポーネント570とLRM機能サブ・コンポーネント576との機能アーキテクチャが、各SLEEと関連づけられたNTおよびLRMサブ・コンポーネントとともに図示されている。図9は、特に、それぞれが各SLEEコンポーネント450および450'と各NT機能サブ・コンポーネント570および570' および各LRM機能サブ・コンポーネント575および575' とを含む各NOSコンポーネント700および700' とをインプリメントする少なくとも2つのコンピューティング・システム440および440' を有する単一NGINサービス・ノードまたは「サイト」45の例を図示する。個別のコンピュータ上で実行する単一のSLEEが図示されているが、2つ以上のSLEEが単一サイトにおいて同一コンピュータ上で動作できる。各SLEE450、450' 上で動作しているのは、S1、・・・S4とされる複数のオブジェクトまたはプロセスで、SLP、LLP、CLP、ELP、永続的に動作しているFD論理プログラムおよびNOSクライアント・オブジェクト558、または他のプロセスの場合がある。

#### 【0115】

ここで説明するように、各NOS NT機能サブ・コンポーネント570および570' は、使用されるデータ・オブジェクトまたはサービス・オブジェクトの正しいバージョンを識別するプロセスを含み、特に、他のプロセス上でプロセスが呼び出すことを可能とし、呼び出されたプロセスの異なるバージョンおよび

インスタンス全体において変化しないままの単一の共通論理名を使用することによる。したがって、NOS NTコンポーネント570は、プロセスからのインスタンスのオブジェクト参照、バージョンング、物理的場所をカプセル化する。

#### 【0116】

ここで説明されるように、各サービス・ノードにあるNOS700の各ローカル・リソース・マネジャー（「LRM」）コンポーネント575、575'は、サービス・プロファイル（コンフィギュレーション）ファイル580中に含まれるコンフィギュレーション・ルールにしたがって、ノードのどのSLEE上でどのサービスを実行するかを決定する。サービス・プロファイル（コンフィギュレーション）ファイル580は、SAから配置されローカルLRMキャッシュ中に格納されるサービス・プロファイルのコンテンツを含んでもよい。LRMは、まず、当該ノードにあるローカル・キャッシュ中に格納されるこのサービス・プロファイル・ファイル580を読み込み、サービス・プロファイル・ファイル中のルールにしたがって、どの特定のSLEE上でサービスを実行するか、および、SLEE中でどのサービスをアクティブに（永続オブジェクトとして）動作させるかまたは要求に応じてのみインスタンス化されるかを決定する。

#### 【0117】

好適な実施形態において、LRM575は、各サービス制御リソースの安定度および状態を追跡することによって、サービス実行の動作時間コンフィギュレーションおよび最適化を可能とする。特に、各LRM機能サブ・コンポーネントは、当該SLEE上で動作するようにプログラミングされた全サービスのリストと、どのサービス・プロセス（オブジェクト参照）がSLEE上でアクティブに動作しているかと、所定の閾値にしたがって当該ノードでのSLEEの現在の負荷状態（処理容量）とを保持する。

#### 【0118】

さらに詳細に記せば、SLEE（サーバ）LRMコンポーネント575は、システム中の各オブジェクト（論理プログラム）に対応するオブジェクト参照のローカル・キャッシュに内蔵された1組のライブラリで、そのオブジェクト参照は、IPアドレスおよびポート番号など、サーバに関する情報を含み、通信を可能

(63)

とする。新規のオブジェクトがシステム内で使用可能となると、それらはNOSに登録される。すなわち、オブジェクト参照はそれらのために作成され、データ管理によってローカル・キャッシュに登録される。

#### 【0119】

そのサービス・プロファイル（コンフィギュレーション）ファイル580に問合せを行って、どのサービスが即座にインスタンス化されるのかを決定した後、NOS LRMコンポーネント575は、サービス・アクティブ化要求を、NOS NT570からSLEEにあるアクティブ・サービス・マネジャー・オブジェクト554へ、同様にSLEE中で実行しているNOSクライアント・インスタンス558を介して送信する。SMオブジェクト554は、SLEEサービスの制御を可能とするAPIオブジェクトである。たとえば、非アクティブ・サービスに対する要求が受信された場合に新規のサービスをインスタンス化する機能を提供する。すなわち、インスタンス化されるときにオブジェクトにプロセス・スレッドを割り当てることが可能で、サービスはLRM575を介してNOSに自分自身を登録する。サービスが、その論理名を使用して別のサービスによって呼び出されると、LRMはコンフィギュレーション・ファイル中のルールを使用して、どのインスタンスが呼び出されるかを、その論理名をアクティブ・インスタンスの物理アドレスに対応づけるためにORB名前サービスを使用することによって決定する。

#### 【0120】

図9に図示するように、NGINサイトまたはサービス・ノード45と関連づけられているのは、個別コンピュータ440' '上、またはコンピュータ440またはコンピュータ440' 'などの共有コンピュータ上のNOSコンポーネント700' '上で動作するサイトLRM577である。サイトLRM577は、1) 各SLEE上で動作する全プロセスの現在の負荷の関数である、各SLEEでのサービスの使用可能性を追跡し；さらに2) 各個別SLEE LRM575のアクティブに更新されるコピーであるリソース・ステータス・リストを、各リソースに対するSLEE識別子とともに維持するように機能する。サイトLRMサブ・コンポーネント577は、複数の評価基準のいずれかに基づいて要求される

サービスのどのインスタンスが使用されるべきかを決定する。複数の評価基準は、それに限定されるものではないが、1) 呼び出しているサービス・インスタンスと呼び出されているサービス・インスタンスとの近似度 (S L E Eが同一か否か、サイトが同一か否か) ; 2) 呼び出されたサービスによって必要とされるデータ管理データと呼び出されたサービス・インスタンスとの近似度 ; および3) 現在のシステムおよびプロセス負荷とを含む。

#### 【0121】

一例として、図9に図示されるように、プロセス (たとえばS L E E 1中のS 1) はS L P、S 4をインスタンス化して特定のプロセス (たとえばV n e t サービス) を実行する必要があるときは、常に、N O Sはまずサービス (すなわちそのオブジェクト参照) がローカル・キャッシュ (たとえばS L E E 1中) で使用可能であるか否かに関する決定を行う。ローカルL R M 5 7 5が要求されたオブジェクト参照を有さない場合、N O Sはサイト・レベルL R M 5 7 7を探して、要求されたサービスに対応する当該オブジェクト参照に場所を決定する。たとえば、図9に示すように、当該オブジェクトはS L E E 2で発見可能であり、発見されると、S L E Eがそれを行うための容量を有していれば、すなわちその使用閾値にまで到達していなければ、当該オブジェクトのインスタンスをインスタンス化することによって当該サービスを使用可能とする。

#### 【0122】

さらに図10に図示するように、各S L E Eに対する各L R M 5 7 7および各サイトに対するL R M 5 7 7に加えて、N O Sコンポーネント7 0 0は、さらに、ネットワーク全体リソース管理機能を実行するプロセスであるネットワーク・リソース・ステータス (「N R S」) サブ・コンポーネント5 9 0を含む。特に、N R Sは、ネットワーク中の各サイトL R M (たとえば図10の各サイト4 4 0 a ~ 4 4 0 cに対応するサイトL R M 5 7 7 a ~ 5 7 7 c) によって保持されるデータのサブセットを含む。N R S 5 9 0は、1) S L E Eのリスト ; 2) どの種類のサービスが各S L E E上で動作するようにプログラミングされているか ; および3) どのサービスが各S L E E上でアクティブに動作しているかを、すなわちパーセント・ベースでのS L E Eの現在の負荷を含む。このN R Sサブ・

(65)

コンポーネント590は、サイトLRM577a~577cが満足できない要求に対する別のレベルの伝播をNOSに与える論理的一括化機能である。さらに、NRSサブ・コンポーネント590は、各SLEE450に対するバイナリ・インジケータを含み、SLEEが上向きか下向きか、およびサービス使用閾値は当該SLEEによって到達されたかを示す。その「上向き」または「下向き」インジケータおよび使用閾値アプリケーションは、SLEEが他のサービスからのサービス要求を受けるために使用可能かを決定するために使用され、これらのインジケータおよび閾値の適用があるとすれば、NRSサブ・コンポーネントは、SLEEが使用可能であるか否かのバイナリ・インジケータを単純に提供できる。一例として、被要求SLPオブジェクトがSLEE中にあるが、そのSLEEが要求されたプロセスをインスタンス化する容量がない場合、当該SLEEに対する使用閾値が到達され、当該サービスに対するさらなる要求を処理できない旨を、サイトLRM577に対して通知を送信する。また、この情報は、NRSコンポーネント590に伝播する(図10)。

#### 【0123】

図8を参照すると、NGINシステムは、監視機構595をインプリメントして、メモリ容量、データベース容量、被要求オブジェクトに対するキューの長さ、キューでの時間量、およびシステム中の各SLEEに対する他のリソース/負荷パラメータを監視する。これらの要素は、これらの要素の1つ以上に基づくSLEEの使用閾値に関する決定を行うNOS700に対して使用可能となる。固定閾値に加えて、複数の閾値がヒステリシスのために使用可能である。

#### 【0124】

では、NOS700が場所独立型処理およびプラットフォーム独立型処理を提供可能としNGINの全体的な処理機能を最適化するNT、LRMおよびNRSを含みNOSによって実行されるリソース管理機能の説明上の例は、図11(a)~図11(b)を参照してさらに詳細に説明される。図11(a)および図11(b)を参照して説明されるLRMプロセス・フロー801において、サービス制御サーバ1上のSLEE1上で実行するサービスS1は、工程802で示すように、サービスS2を呼び出す必要がある。サービスS1は、スイッチ構造呼

(66)

制御からイベント・サービス制御要求を受信したFDまたはサービス論理プログラムであり、たとえば呼処理を完了するために、別のSLP、S2を呼び出す必要がある。

#### 【0125】

特に、図11(a)を参照すると、サービスS1は、SLP S2に対する論理名を使用してNOS700へ要求を発行する。サービス・オブジェクトに対するSLP要求を受信されると、NOS名前変換機能570aは、工程804に示されるように実行され、NOSは被要求サービスがローカル・サービス制御サーバ1上でアクティブに動作することを認識している、すなわち、被要求サービスの論理名と関連づけられたオブジェクト参照を有するかを決定する。ローカル・サーバ・キャッシュ中に格納されるデータは、以下のNOSネーミング・データ・フィールドを含むことが好ましい。1) 典型的に、サービスを説明する論理名であり機能ディスクリミネータ・データが指す名前であるSLP論理サービス名；2) たとえばサービスが動作する当該バージョンを要求する特定の顧客またはノードなどに対して、必要とされる場合のある特定のサービスのバージョンを説明する任意のバージョン番号；3) 配置（すなわち、SAがノードにワーク・パッケージを配置したがサービスはアクティブ化されていない場合）、アクティブ（すなわちサービスが現在アクティブであることを示す）、またはフォールバック（たとえば高速の反転を実現するために、サービス・オブジェクトの前のバージョンにもどることが所望された場合）を含むステータス；4) オブジェクト・インスタンスの物理的場所を識別するIPアドレス、ポート、およびその他の情報を含むことが可能なオブジェクト名または参照；5) 稼動日時および休止日時；6) たとえばオブジェクトが使用可能でない、またはアクティブ化不可である場合のエラー・プロセス・オブジェクト名；および7) フォールバック・ステータスにあるときに実行されるフォールバック・オブジェクト名。ローカル・サーバNOSネーミング・プロセスは、LRMステータス・プロセッサ（不図示）によって提供されるサービスを利用する。LRMステータス・プロセッサは、ローカル・サーバ・キャッシュ・ステータス・データベースを、サービス制御サーバ中の特定のSLEE中で動作している現在アクティブなサービスに関してのみ更

(67)

新する。したがって、ローカル・サーバNOS名前変換機能は、まず、ローカルに実行されることが可能である。NOSがまず名前要求を得ると、論理名を探してオブジェクト名（またはオブジェクト参照）を獲得する。NOSは、論理名からオブジェクト名を獲得して、ノードLRMプロセスは、工程806で示すように、1つ以上のビジネス・ルールに基づいて処理される、被要求オブジェクトの最良のインスタンスを決定する。

#### 【0126】

工程804において、論理名が認識され、オブジェクト参照が使用可能である場合、処理は、工程806のLRM機能に進み、使用閾値など、特定の評価基準にしたがって、SLEE1上で動作するS2のアクティブ（「使用可能」）なインスタンスを決定する。アクティブなインスタンスがまったく発見されない場合、LRMはS2がSLEE1上でプログラミングされているか、それにもかかわらずインスタンス化されていないかを確認するために調査可能である。そうである場合、SLEE1が十分な使用可能容量があれば、NOS700は、SLEE上のS2のインスタンスをインスタンス化することを決定可能である。上述のように、サーバ・レベルのLRMは、サーバでアクティブなものを意識し、インスタンス化されたものを意識するのみである。オブジェクトが現在アクティブで、ローカル・サーバ・レベルでインスタンス化される場合、このサービスのための新規のスレッドをインスタンス化するオブジェクト参照がSLP要求に返される。NOSは新規のサービス・スレッドのインスタンス化を開始し、返されたオブジェクト参照に基づいて要求されたサービスを実行し、まだインスタンス化されていない場合、オブジェクト参照を返す。

#### 【0127】

工程804において、SLEE1が十分な使用可能容量を有していない場合、またはS2がSLEE1上での動作のために使用可能でない場合、工程810において、SLEE1上のLRMは、サイトLRM577aにサービス要求を送信する（図10）。サイトLRMは、同様のビジネス・ルールを適用し、当該サイトの別のSLEE上で、S2のインスタンスがアクティブであるか、またはインスタンス化すべきかを決定する。したがって、工程810において、ノードN

○S名変換機能はインプリメントされ、要求された論理名が当該ノードで使用可能であるか、すなわち当該ノードで同一または異なるローカル・サービス制御サーバで別のSLEEが要求されたサービスの論理名と関連づけられたオブジェクト参照を保持するか否かを決定する。工程810において、その論理サービス名が認識される場合、NTサブ・コンポーネント570は、NOS LRM575に問合せを行い、S2のどのインスタンスを使用するかを決定する。その後、工程814において、ノードLRMはノード・キャッシュ・ステータス・データベース（不図示）に対してビジネス・ルールを適用し、アクティブであれば、要求されたサービスに対する所望のオブジェクト参照を検索しそのアドレスを呼び出ししているSLPに返す（図11（a）工程802）。サービスが現在インスタンス化されていない、または特定のSLEE上の要求されたサービスが処理負荷または課せられている他の制約に起因してインスタンス化されないことを決定した場合、工程818において、割り当ておよびローディング処理が実行される。これは、ノード・キャッシュ・ステータスデータベースを調査し、たとえばサービス近似度、データ近似度、閾値、現在の処理負荷などに関する該当ビジネスルールをインプリメントすることによって行われ、サービス・オブジェクトがインスタンス化可能であることを決定した場合にSLEE中で要求されたサービスをインスタンス化し、そのアドレスを呼び出ししているSLPに返す。なお、SLEE毎のインスタンス化に対して1つより多いサービスが使用可能である場合、どのサービス・スレッドがインスタンス化されるかを決定するには、ラウンド・ロビン方式が実行可能である。

#### 【0128】

図11（a）に戻って参照すると、工程810において、現在のノードが要求された論理名を認識しない場合、すなわちノード・キャッシュが要求されたサービスの論理名と関連づけられたオブジェクト参照を有さない、または適用されたビジネス・ルールに起因して、ノードでオブジェクトをインスタンス化できない場合、工程822において、グローバル・ネットワーク・リソース・ステータス（NRS）プロセス590に対して問合せが行われ、知的ネットワーク170全体におけるSLEEの現在のステータスを調査し、S2に対するサービス要求を



処理可能である S L E E を決定する。この前に、工程 8 2 0 において示すように、調査が行われ、オブジェクト参照を発見するためにネットワーク・リソース管理に対して問合せが行われた回数を表す索引番号が所定の限界（たとえば 3 回）を超過したか否かを決定する。この閾値が超過された場合、プロセスは終了し、アドミニストレータはサービス・オブジェクトが発見不可であることと、工程 8 2 1 に示すように、エラー状態が存在することとを通知される。N R S 問合せ閾値が超過されていない場合、工程 8 2 2 に示すように、N R S プロセス 5 9 0 は、ネットワーク中のどのサービス・ノードが要求されたサービスを実行可能であるかを決定する。知的ネットワーク中のノードを決定後、プロセスは図 1 1 (b) の工程 8 2 4 に続き、ノード N O S 名前変換機能 5 7 0 b は、要求されたサービスの論理名と関連づけられたオブジェクト参照を獲得するためにインプリメントされる。工程 8 2 4 において、当該ノードでその論理サービス名が認識されない場合、工程 8 2 9 において、N R S 問合せ索引番号はインクリメントされ、プロセスは、図 1 1 (a) の工程 8 2 0 に戻って進み、索引番号閾値が超過しエラー状態が存在するかを調査する。図 1 1 (a) 工程 8 2 0 において、N R S 問合せ索引が所定の閾値を超過していない場合、工程 8 2 2 において、N R S プロセス 5 9 0 に対して再度問合せが行われ、別のサービス・ノードにある使用可能サービスの新規の場所を発見する。

#### 【 0 1 2 9 】

ステップ 8 2 4 において論理ネームが認識されると、そこで処理はステップ 8 2 6 へと続き、受容可能な処理負荷に従って要求されたオブジェクトリファレンスに関連するアドレスを決定する。このアドレスは、そこで図 1 1 (a) のステップ 8 0 2 に示すように要求した S L P に対し送り返される。ステップ 8 2 6 において、そのサービスがそれまでインスタンス生成（アクティブ）されていない場合は、処理はステップ 8 2 8 へ進み、そのノードにおけるノードキャッシュ状態データベース 7 6 8 をチェックし、ビジネス規則を実行し、サービスオブジェクトがインスタンス生成に利用できると判定された箇所の S L E E における要求サービスをインスタンス生成することにより、割当と負荷処理とを可能にする。続いて、インスタンス生成されたオブジェクト S L P のアドレスがステップ 8 2

4において要求クライアントへ送り返される。

【0130】

アクティブなインスタンスS2が選択されると、そのS2インスタンス生成用のオブジェクトリファレンスがSLEE1上のNTへ送り返される（ステップ802）。そして、NTは論理ネームS2を選択されたインスタンスS2用オブジェクト識別子へと効率的に変換し、そのS2用オブジェクト識別子をS1とS2との間の相互処理通信に用いる。オブジェクト識別子は、IPアドレスと、ポートと、オブジェクトインスタンスの物理位置を特定する他の情報とを含む。オブジェクトリファレンスが決定されたならば、そこでNOSはCORBA準拠ORB及び、UDP/IPのようなプロトコルを除くデータ通信接続を実行することにより、二つのサービス間でのオブジェクト連結性をもたらす。同じSLEE上で稼働中であろうと或いは数千マイル彼方の別のサイトの他のSLEE上で稼働中であろうと、呼出しを受けたサービス位置は電話サービスにとっては完全に明らかである。かくして、呼出しを必要とするSLPが遠隔サイトのSLEE上でインスタンス生成されると、この呼出しはこれを受信したスイッチ上に依然として保持される。好ましくは、オブジェクトリファレンスが一度でもアクセスされた場合、例えばNRSレベルを経由する他のサイトでは、NBSがサービス管理プログラムを介してオブジェクトリファレンスが将来のリファレンス用に要求サイトにてキャッシュされ聴取されることを保証するとよい。かくして、現在の例では、このサービスが再び必要とされるときに、サイトLRMのテーブル探索を起動することによる一連のテーブル探索を低減するため、サービスS2用のオブジェクトリファレンスは、場所の如何によらず、その後にSLEE1のLRM575のローカルキャッシュ内にキャッシュされる。SLEEにおいてサービスオブジェクトリファレンスが供給される仕方が様々であることは、当業者には明白なことである。例えば、NOSデータ模写機構は、サイトLRM577における全てのオブジェクトリファレンスをそのサイトにおいて各SLEE用に個別LRMに模写するよう設けられる。

【0131】

1-800番呼出し（「18C」）のコンテキストでは、18C呼出し処理と

(71)

サービス利用シナリオを、図13 (a) 乃至13 (c) のフローチャート及び図18の概略機能線図を参照し、例示目的に以下に説明する。最初に、ステップ920に示すように、呼出しは先ずNGSスイッチ機構75に届く。NGSが呼出しを受けると、ベアラ制御要素(図5)が、呼出しを受けたアクセス回線に対し、ANIやダイヤル番号や呼出し処理に必要な他のデータ同様、呼出し制御要素を供給する。呼出し制御要素は、プログラムされたロジックに従って実行されるのに伴い、呼出しのための状態モデルを維持する。状態モデルにはさらに、ELP540をインスタンス生成して図18に示すFD510に対しサービス要求を送信するトリガが含まれる。ELPをインスタンス生成するため、NGS呼出し制御要素90は、図13 (a) のステップ923に示したように、ELP用の論理ネームを用いて、NOSへのメッセージをアドレス指定する。NOSはこれに応答し、サービス管理プログラムオブジェクト(図8)に対しメッセージを送信し、SLEE内でELPをインスタンス生成し、ステップ926に表したように、そのELP向けのオブジェクトリファレンスを呼出し制御機構へと返送する。NGS呼出し制御要素は、ステップ929に示す如くSLEE内のFDに対し送信されるサービス要求メッセージ内に、オブジェクトリファレンスを含む。かくして、どんな処理によっても呼出しに対し生成された全ての適格事象データが、インスタンス生成されたELPプロセスへ書き込まれる。特に、サービス要求メッセージはFD用論理ネームに対しアドレス指定され、呼出しを受信したのと同じサービスノードにおいて、稼働中のFDロジックプログラムのための物理アドレスへNOS・NT要素により変換される。サービス要求メッセージには、ダイヤルされた番号とANIと他のデータとが含まれる。

### 【0132】

次に、ステップ931に示すように、FDは機能識別テーブルを用い、どのSLPが受信サービス要求を処理しようとしているかを識別する。例えば、受信メッセージが18Cサービス要求である場合、18C・SLPにより処理すべきとなる。下記のテーブル3は、例えば1-800番といった様々な無料電話サービスへのポインタを含む見出しを有する略記FDテーブルの一例である。

エントリーポートテーブル

(72)

「001001」 SLPポインタ 「Vネット」

「001001」 FGDテーブルへのテーブルポインタ

FGDテーブル

1800★テーブルポインタ 800 テーブル

1888★テーブルポインタ 800 テーブル

1900★テーブルポインタ 900 テーブル

1★SLPポインタ「市内局番」

800テーブル

「1-800-C」への1800コレクトSLPポインタ

18008888000SLPポインタ「800サービス」

1800★ SLPポインタ「800サービス」

1888★ SLPポインタ「800サービス」

ここで、FGDは機能グループ識別子である。

【0133】

特に、回線網（交換台）内で呼出しが発生した箇所及び受信された呼出し種別（例えば、1-800番）に基づき、FDは適当なSLP論理ネームを決定する。例えば、識別番号「001002」は、FGDテーブル（FGDテーブルへのポインタ）内のテーブル探索を要求する呼出しの領収書を示す。FGDテーブルの方は、呼出し番号例えば800★に基づき、他のテーブルへのポインタを維持する。ただし、「★」は区切り文字である。この800テーブルからは、例えば、FDは要求されたSLP論理ネームへのポインタを入手する。その結果、このSLPが呼出され、サービス要求がNOSに渡され、NOSが要求のあった18Cサービスに従ってCLP545、LLPO530及びSLP520のオブジェクトをインスタンス生成する。例えば、LLPOについては、LLPO用の論理ネームが呼出しを受信したベアラ制御回線に基づいてNOSへ供給される。この回線の識別は、ベアラ制御要素80により識別されたANIもしくはアクセス回線のいずれかに基づくものである。ANIは呼出しを発した原アクセス回線を識別するが、それはNGSが呼出しを受けたアクセス回線と同じだったり異なったりもする。すなわち、受信された呼出しが、例えばローカルネットワーク上で発

生したもので、互換キャリアネットワーク上のスイッチ75を通過したものであることがある。それ故、呼出し待機すなわちお話中といった回線関連機能は、ANNIによって識別することができる。NOSは、LLPOのインスタンス生成用にLLPO用の論理ネームを物理アドレスへ変換する。他の論理プログラム（SLP等）は他のサイトにおいてインスタンス生成されるが、一方でLLPをその関連回線の在るサイトにおいてインスタンス生成することもできる。LLPはSLEEにおいてインスタンス生成されるが、SLEEはサービス制御サーバもしくは呼出し制御サーバ上にある。インスタンス生成がなされると、回線関連機能用のデータ管理機構に照会し、発生回線の状態を維持し、呼出し待機やオーバフローによる経路指定などのあらゆる機能を、これらの機能が呼出し元（例えば、呼出し待機中）や回線網（オーバフローによる経路指定中）によって呼出されたときに、呼出す。

#### 【0134】

図13（a）のステップ934を参照するに、NOSは例えば18Cといった呼出される特定のサービスを表す論理ネームを含む機能識別子からサービス要求の引き渡し要求を受信する。NOSは、その要求がインスタンステーブル（図示せず）内の論理ネームならびに表を含むことを確認し、このサービス要求に応える可能ないかなるSLPプロセスを有するか否かを判定する。NOSはまた、ステップ937に示すように、要求された種別のどのインスタンスを使用すべきかをNOS・LRM機能を通じて確認する。そして、ステップ941に示すように、NOSはサービス制御SLEE上で稼働中のサービス管理プログラムオブジェクトに対し要求を送信し、要求された18C・SLPサービスを呼出す。好ましい実施形態では、NOSはNGSから入来する原サービス要求通告を受信したサービス制御サーバからのSLPを選択するが、NOS・LRMの実行を通じかつサービス制御インスタンスのリスト及びそれらの現在の状態とに基づいて、NOSがあらゆるサービス制御要素内のSLPを選択し得ることが分かる。ステップ943に示すように、NOSは選択されたSLPが既にインスタンス生成されたか否か、そして選択されたSLPが未だインスタンス生成されていないかどうかを判定し、ステップ946に示すように、SLPオブジェクトをインスタンス生

成するようSMに指示する。さもなくば、選択されたSLPが既にインスタンス生成されている場合は、スレッド管理プログラムが、ステップ945に示すように、SLPオブジェクトに対し新規の処理スレッドを割り当てる。

#### 【0135】

図13(b)の次のステップ949は、インスタンス生成されたSLP処理がその物理アドレスをNOSに登録することと、NOSがそのSLPをサービス要求に割り当てることを要求する。そして、NOSは、ステップ951に示すように、新規のSLPに対しサービス要求引き渡しメッセージを渡す。それと並行して、NOSはインスタンス生成されCLPに対し、インスタンス生成されたSLP、ELP及びLLPOのオブジェクト用のオブジェクトリファレンスを含む全データを送信する。CLPとELP向けのオブジェクトリファレンスはまたLLPOとSLPへと供給され、これによりLLPOとSLPはCLP及びELPとのインターフェースをとることができるようになる。最後に、ステップ954に示すように、ここでSLPはプログラムされたロジックに従って呼出し処理を開始する。

#### 【0136】

18C呼出しのコンテキストでは、18C・SLP520は好ましくは、18C経路指定データベース（図示せず）から必要なデータを入手し、適切な決定を下す。図13(c)に示すように、18C・SLP520は、以下のステップを呼出す。すなわち、ステップ960において、NOS・NT向けに18C経路指定データが必要とする論理ネームを送信し、ステップ962に示したように、18C経路指定データベースの論理ネームをDMに照会し、実際の18C経路指定DBネームとその格納位置とをDMから受信し、ステップ964に示したように、NOS・LRMに対し要求を発して18C経路指定データベースの局所的な利用が可能であるかどうかを判断し、可能であれば、ステップ966に示したように、18Cデータベースの物理アドレスを18CSLPへ送り返し、ステップ968に示したように、呼出された800-何番と回線IDと原スイッチ中継回線とANIとを送信することにより、顧客プロファイルのテーブル探索についてデータ管理機構へ照会を送信し、ステップ970及びステップ972に示すように

、経路指定応答で特定された終端の実際の終端位置（ノード）のテーブル探索をDMに対し要求し、DMから実際の終端ノード位置を受信することにより、18 C・SLPへ戻るスイッチ／中継回線を含む最終の経路指定情報をDMから受信する。その後の処理は、例えばDM内に格納された呼出しコンテキストデータ内に配置するため、ELP510に対する経路指定応答情報の送信を必然的に伴い、経路指定情報を含むCLP545向けの引き渡しコマンドとともに発呼要求を送信する。このシナリオでは終端ノードは遠隔地にあるが、この場合、遠隔ノード上の終端LLPをインスタンス生成し、プロファイルのテーブル探索を実行して終端回線上のあらゆる機能を判定することが必要になる筈である。他のサービスの流れのシナリオでは、SLPは1以上の他のSLPを呼出さばならないこともある。

#### 【0137】

SLPが呼出しに対し回線網の終端を決定するか、或いはさもなくば例えばDTMFディジット又は再生音声を検知するよう資源複合体に対する対応を決定すると、ステップ957に示すように、NGS呼出し制御機構に対しサービス応答メッセージを送信する。呼出し制御機構は、そこでステップ959に示すように、NGSスイッチ75'（図14）に指示して回線網終端への呼出しを準備し完了させることを含む指示を実行する。

#### 【0138】

より詳細には、呼出しダイアルにおいてNOSエージェントに対し進められるLLPO（発生回線）への引き渡しコマンドを伴う発呼を送信し、そのことで呼出しが終端ノードへ経路指定される。ELPプロセスは、そこで、発呼呼出しコンテキストデータをDMへ書き込む。

#### 【0139】

ステップ957へ戻ると、SLPが回線網終端用物理アドレスの呼出し制御へ復帰すると、そこでLLPTプロセス531は呼出しが終端された回線についてインスタンス生成される。このことが、NOSをしてLLPT用の論理ネームをSLPにより与えられた回線網終端に関連付けることを可能にする。ただし、この論理ネームは、SLP（一実施形態）もしくはFD（他の実施形態）へのサー

(76)

ビス要求内の呼出し制御のどちらかによって与えられるものである。NOSの方は、終端回線が存在する箇所のサービスノードにおいてSLEE内のLLPTをインスタンス生成する。

#### 【0140】

或いはまた、ステップ957において、SLPが18C呼出しの処理例における例えばIVR機能といった特定資源に対する要求を代わりに送り返すこともある。NGSは何れの資源、すなわちIVRの可能性をもったスイッチポートとかVRUポート等の何れを割り当てるべきかを決定し、資源用アドレスをSLPに対して送り返す。SLPはそこで、その資源に関連したLLPT用のアドレスを（データ管理機構への照会を介して）識別し、そのLLPTのインスタンス生成を要求する。その呼出しは、そこでその資源へと経路指定され、恐らくNGINに対する他のサービス要求とともに処理される。

#### 【0141】

呼出しが完了すると（すなわち、両者の接続が切断されると）、LLPTは各スイッチ75, 75'（図14）におけるNOS要素から呼出し完了通告を受け、呼出し完了通告をCLPに向けて送り出す。CLPは、関連するLLPS及びELPに対し呼出し完了通告を送り出し、CLP通告によりトリガされるのに伴って強制切断する。切断に先立ち、呼出し完了後に例えば請求と他の様々な目的のための維持する必要のあるELP呼出し詳細データが、先ず格納されることになる。

#### 【0142】

上記のごとく、いくつかの好ましい実施形態を詳細に記述した。本発明の範囲はまた、記述したものとは異なる実施形態であっても、特許請求の範囲にさえ含まれる実施形態であれば包含することは理解すべきである。

#### 【0143】

例えば、汎用コンピュータは、一つの型の応用向けに特別に作られたものでないことが分かる。汎用コンピュータは、発明を実施するのに必要な機能を遂行できるいかなる大きさのどんな計算機であってもよい。

#### 【0144】



追加例としては、「ジャバ」プログラミング言語は、発明の実施に必要な同様の機能を有して同様の機能を遂行する他の等価なプログラミング言語でもって置換可能である。

#### 【0145】

ここにおけるこれらの用語の使用法は、他の用語と同様、発明をこれらの用語だけに限定する意図をもつものではない。使用する用語は、同義語及び／又は等価物を指す他のものと置換することもできる。包含する言葉は、発明の範囲に鑑みて非網羅的に解釈さるべきである。本発明の様々な実施形態が、ハードウェアやソフトウェア或いはマイクロコード化されたファームウェアを採用したり、そこに埋設したりできることも理解さるべきである。

#### 【0146】

本発明は、上述の実施形態に関連して開示しかつ論じてきたが、当業者にとっては、本発明の意図と範囲内において、多数の変更と変形と修正が可能であることは明白である。従って、それ故に、以下の特許請求の範囲は、そうした変形と修正を包含させたものであることを意図したものである。

#### 【図面の簡単な説明】

【図1】 様々なスイッチングアーキテクチャの論理表現である。

【図2】 従来技術による典型的なインテリジェント・ネットワーク・コンフィグレーションを使用する通信システムの図表である。

【図3】 インテリジェント分散ネットワーク・アーキテクチャを使用する通信システムの図表である。

【図4】 (a) は、次世代インテリジェント・ネットワークのSAおよびDMコンポーネントを示すブロック図である。(b) は、サービス管理コンポーネント300の機能を概念的に示す。(c) は、データ管理コンポーネント400の機能アーキテクチャを示す。

【図5】 本発明によるインテリジェント分散ネットワーク・アーキテクチャを使用する通信システムの論理機能図である。

【図6】 本発明によるインテリジェント呼処理装置内の機能インタフェースの積層化を示す図表である。

【図7】 本発明によるインテリジェント呼処理装置の管理対象オブジェクトのクラス階層を示す図表であ。

【図8】 サービス制御環境430の好ましいアーキテクチャを示す。

【図9】 NOS NTおよびLRM機能サブコンポーネントの機能的アーキテクチャを示す。

【図10】 インテリジェント・ネットワークのリソース管理システムのアーキテクチャを示す。

【図11】 (a)は、3層インテリジェント・ネットワーク・リソース管理機能を示す。(b)は、3層インテリジェント・ネットワーク・リソース管理機能を示す。

【図12】 (a)は、SLEE起動プロセスを示す。(b)は、サービス・マネージャ・プロセスを示す。(c)は、SLEEクラスローダ・プロセスを示す。(d)は、サービス・エージェント機能を示すフローチャートを示す。(e)は、サービス・エージェント機能を示すフローチャートを示す。(f)は、スレッド・マネージャ・プロセスを示す。(g)は、サービス・エージェント・ポストイベント・プロセスを示す。

【図13】 (a)は、1-800/8xx呼処理サービスを実行するプロセスフローの例を示す。(b)は、1-800/8xx呼処理サービスを実行するプロセスフローの例を示す。(c)は、1-800/8xx呼処理サービスを実行するプロセスフローの例を示す。

【図14】 IDNA/NGINによって処理される呼処理シナリオを示す。

#### 【符号の説明】

172 ICP

180 リソース複合体(RC)

200 インテリジェント分散ネットワーク・アーキテクチャ

202 広域ネットワーク(WAN)

204 IDNAノード

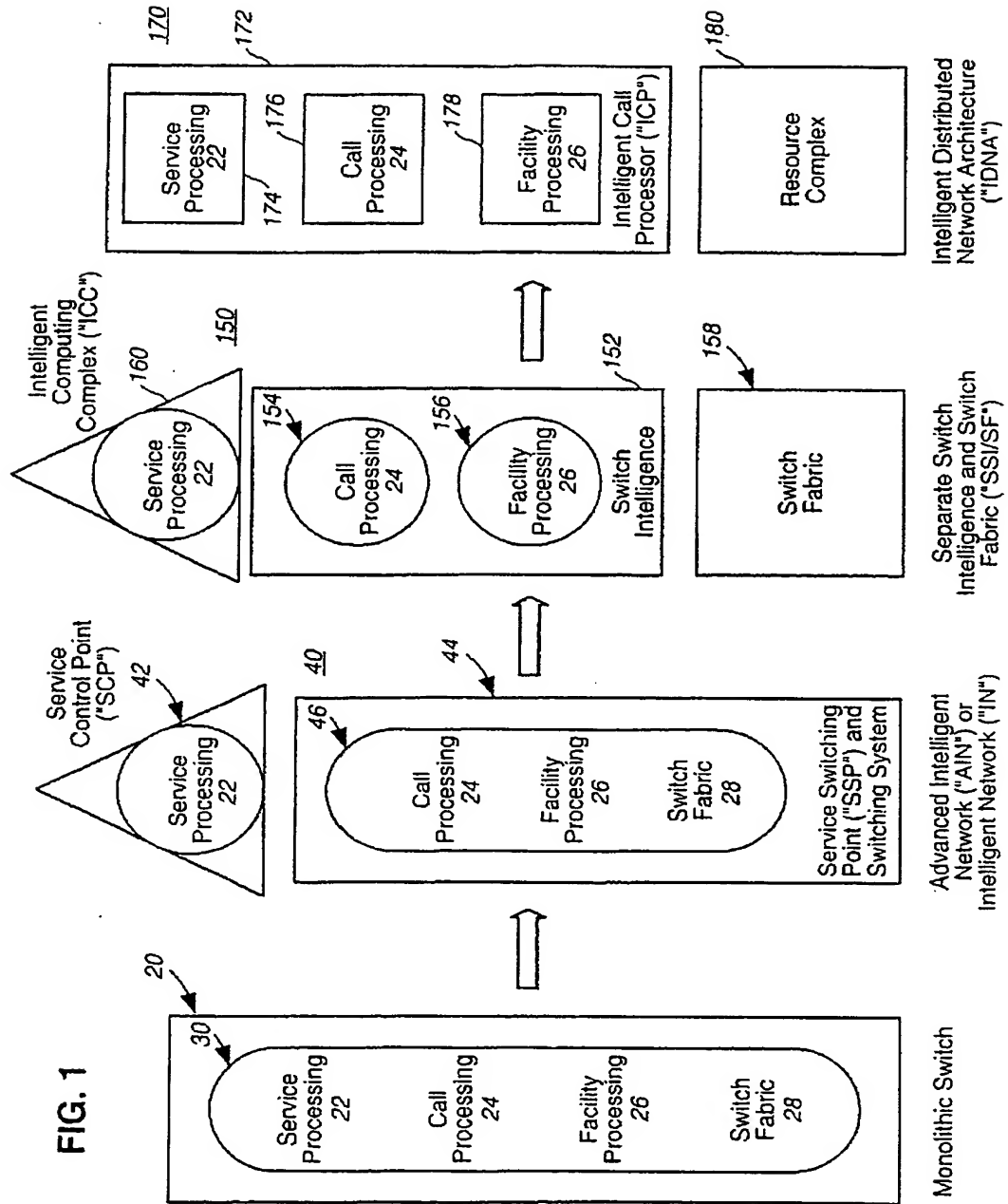
210 補助処理装置

(79)

- 2 1 2 N M S
- 2 1 4 直接リンク
- 2 1 6 信号
- 2 1 8 伝送線
- 2 2 4 高速データ通信パイプ
- 2 2 6 操作リンク
- 2 2 8 M O C E
- 2 3 0 レポジトリ
- 2 4 0 I C P - N M S エージェント
- 2 4 2 S L E E
- 2 4 4 管理対象オブジェクト
- 2 4 6 リソースプロキシクラス
- 2 4 8 支持体制御クラス
- 2 5 0 コール制御クラス
- 2 5 2 サービス制御クラス
- 2 5 4 他のリソース複合体

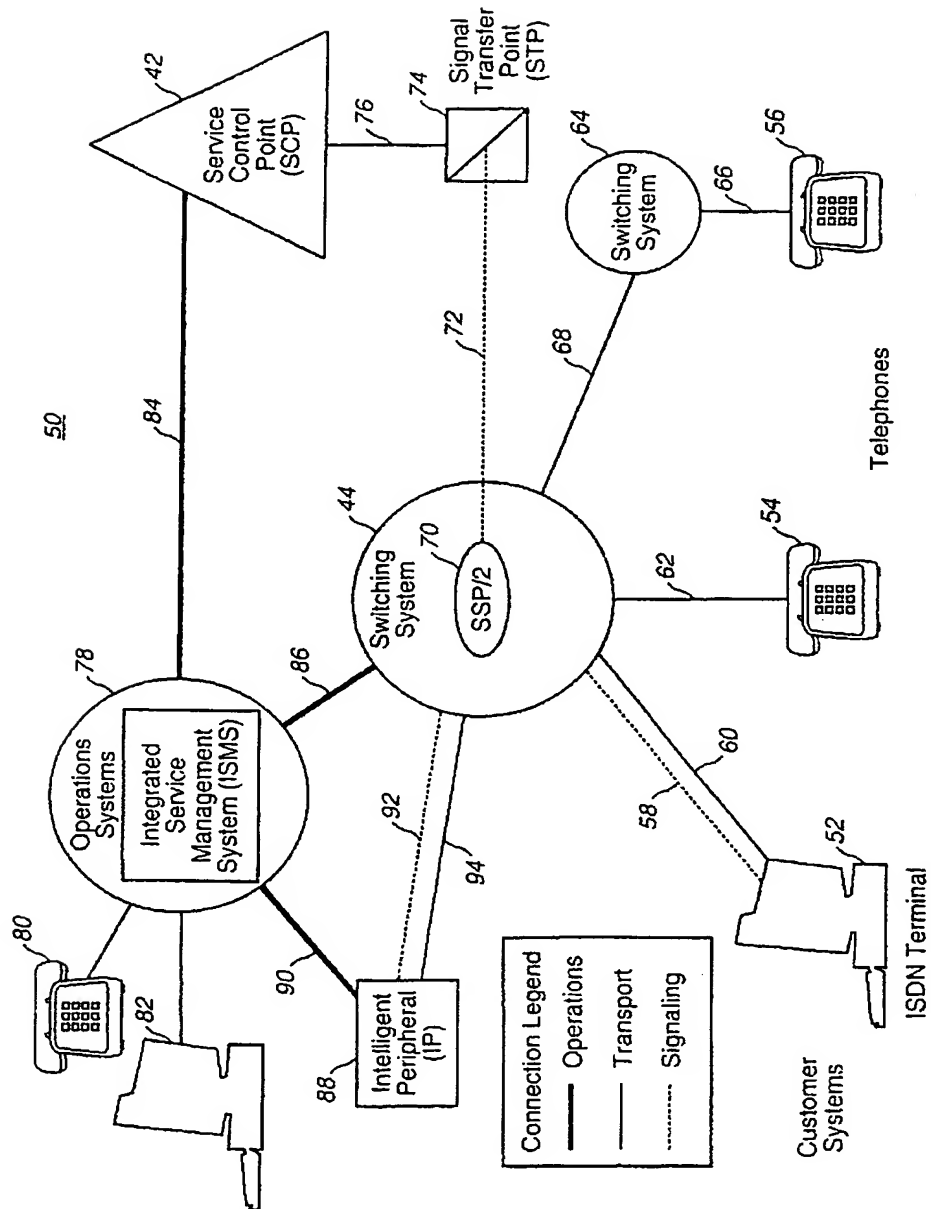
(80)

【図1】



(81)

【図 2】



(Prior Art)  
FIG. 2

(82)

【図 3】

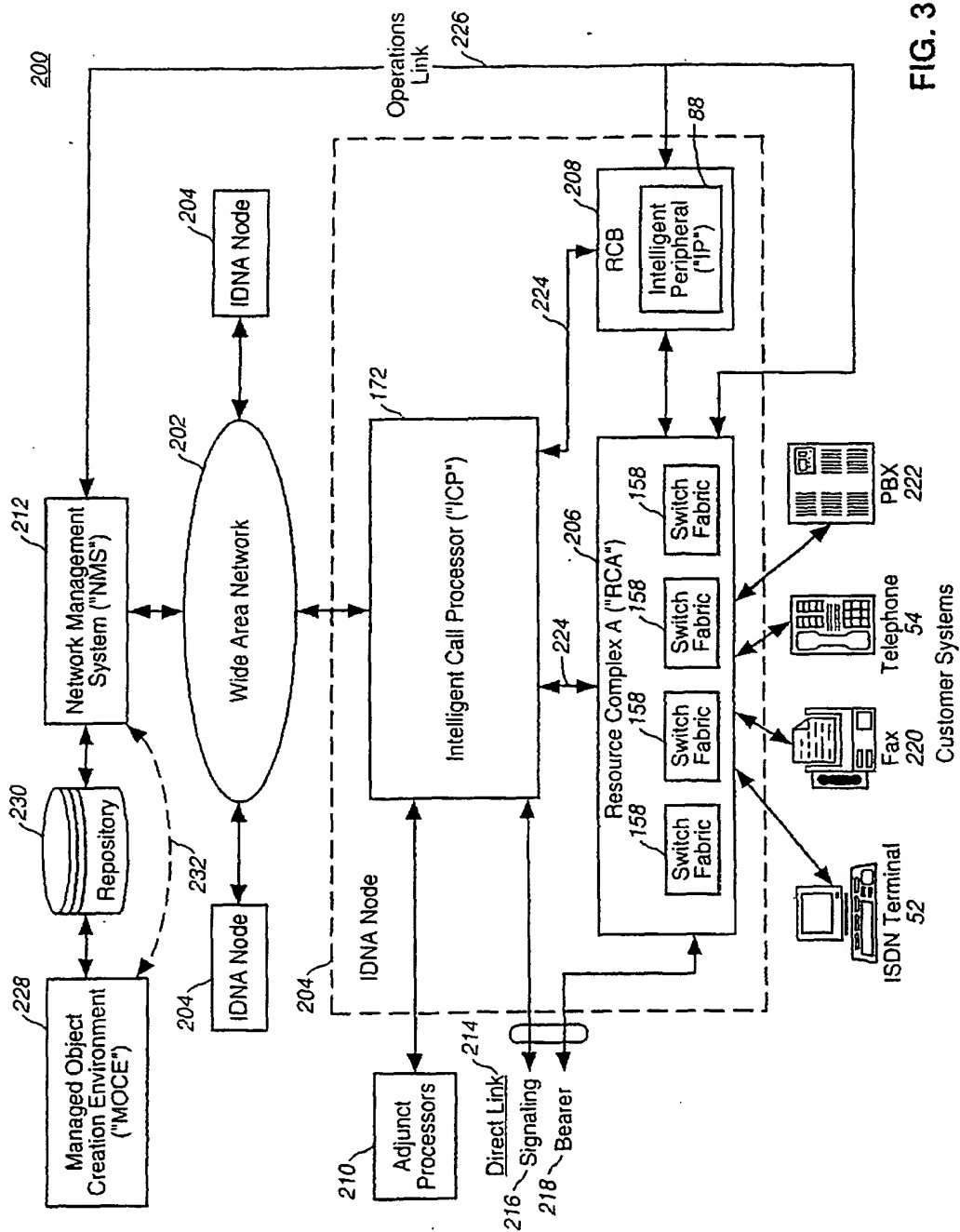


FIG. 3

(83)

【図 4 a】

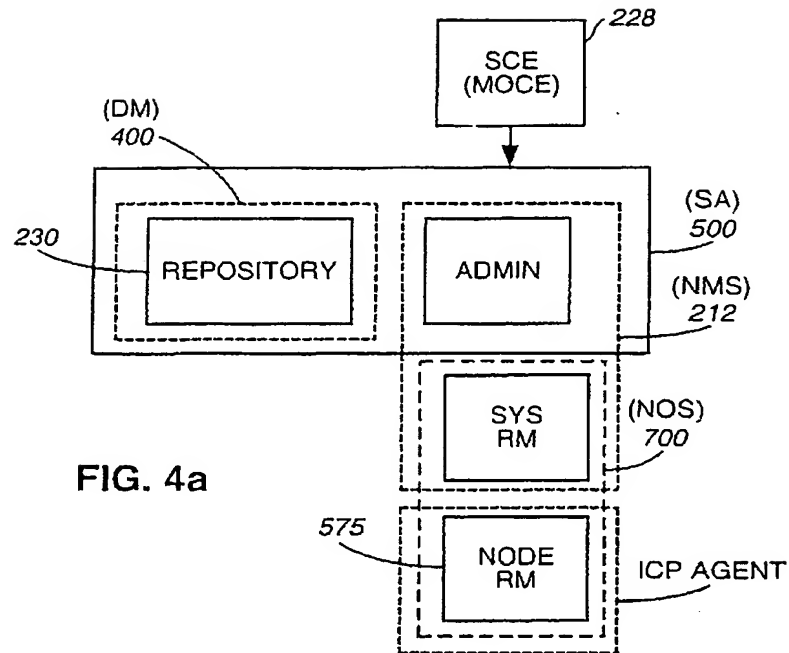


FIG. 4a

【図 4 b】

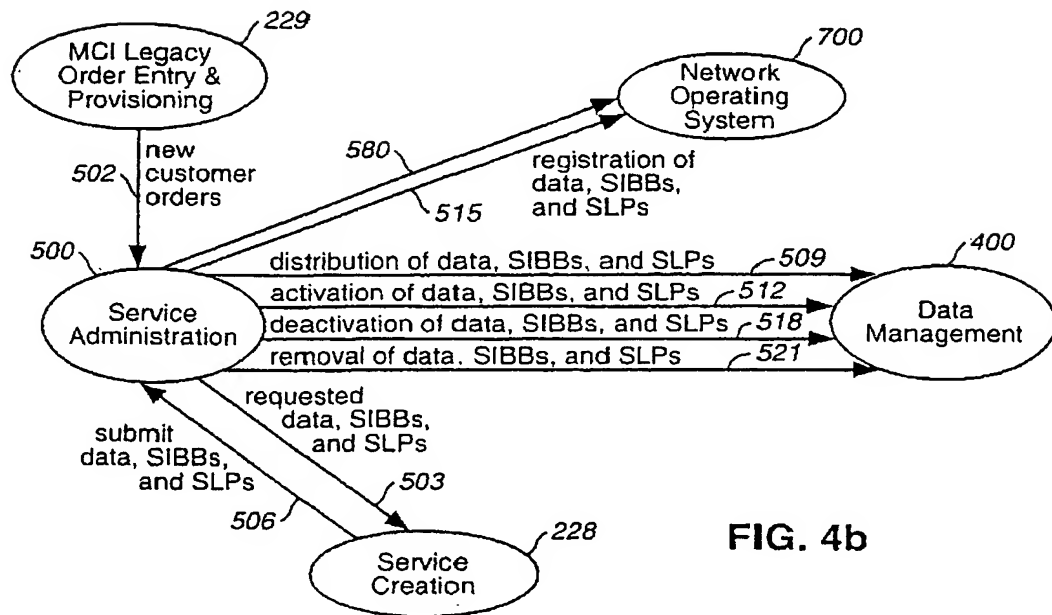


FIG. 4b

(84)

【図 4 c】

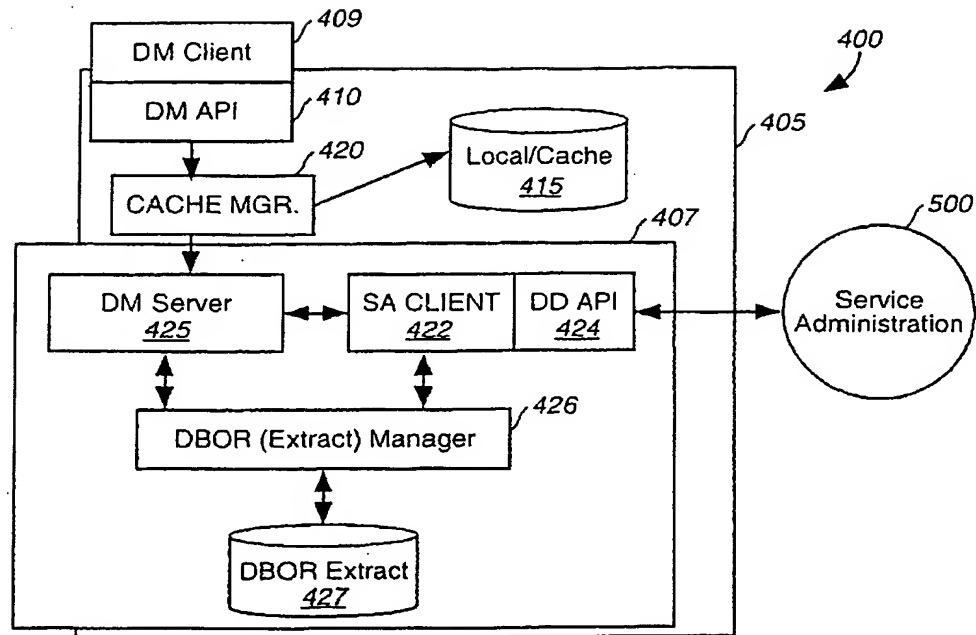


FIG. 4c



(85)

【図 5】

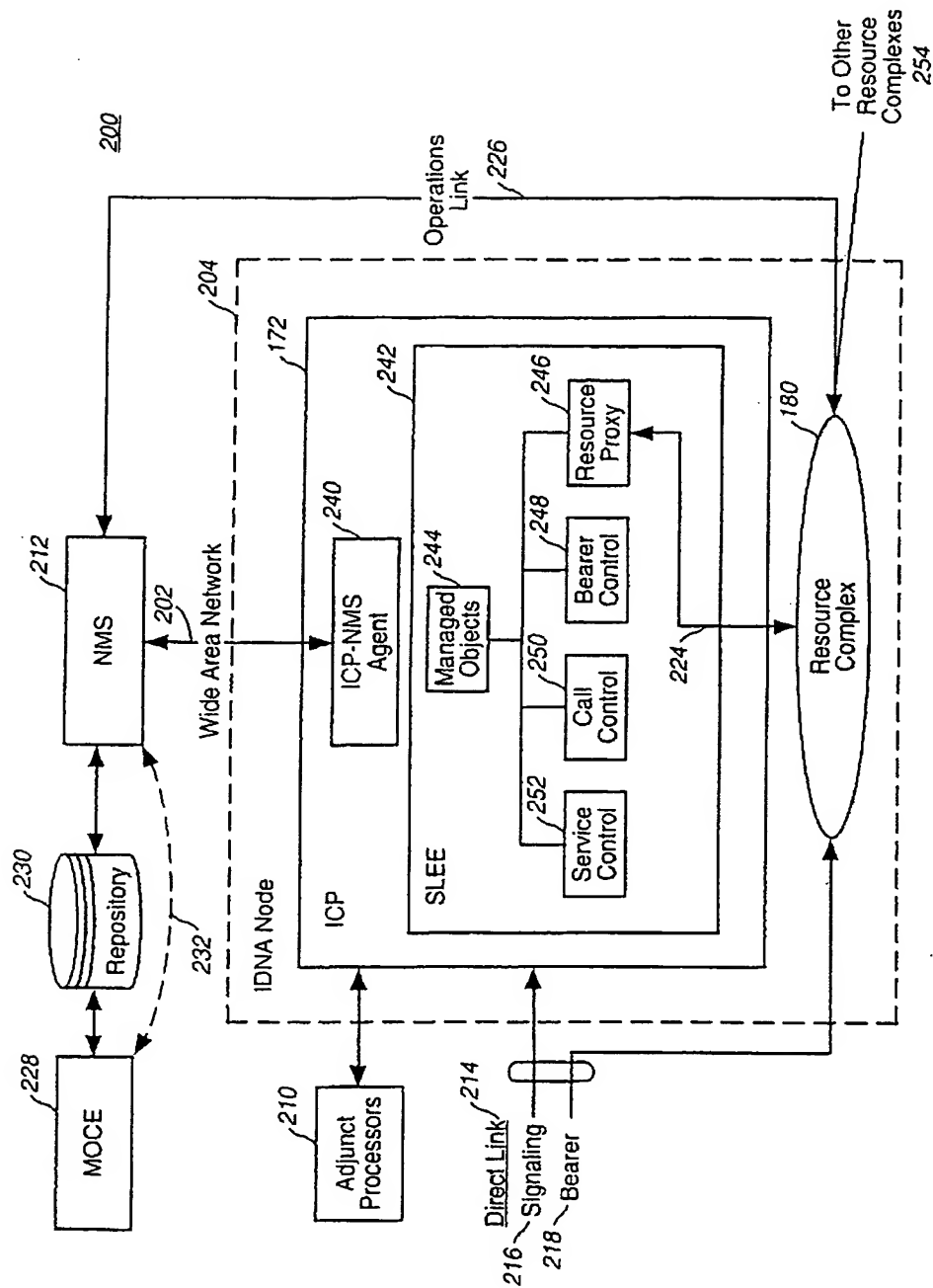


FIG. 5

(86)

【図 6】

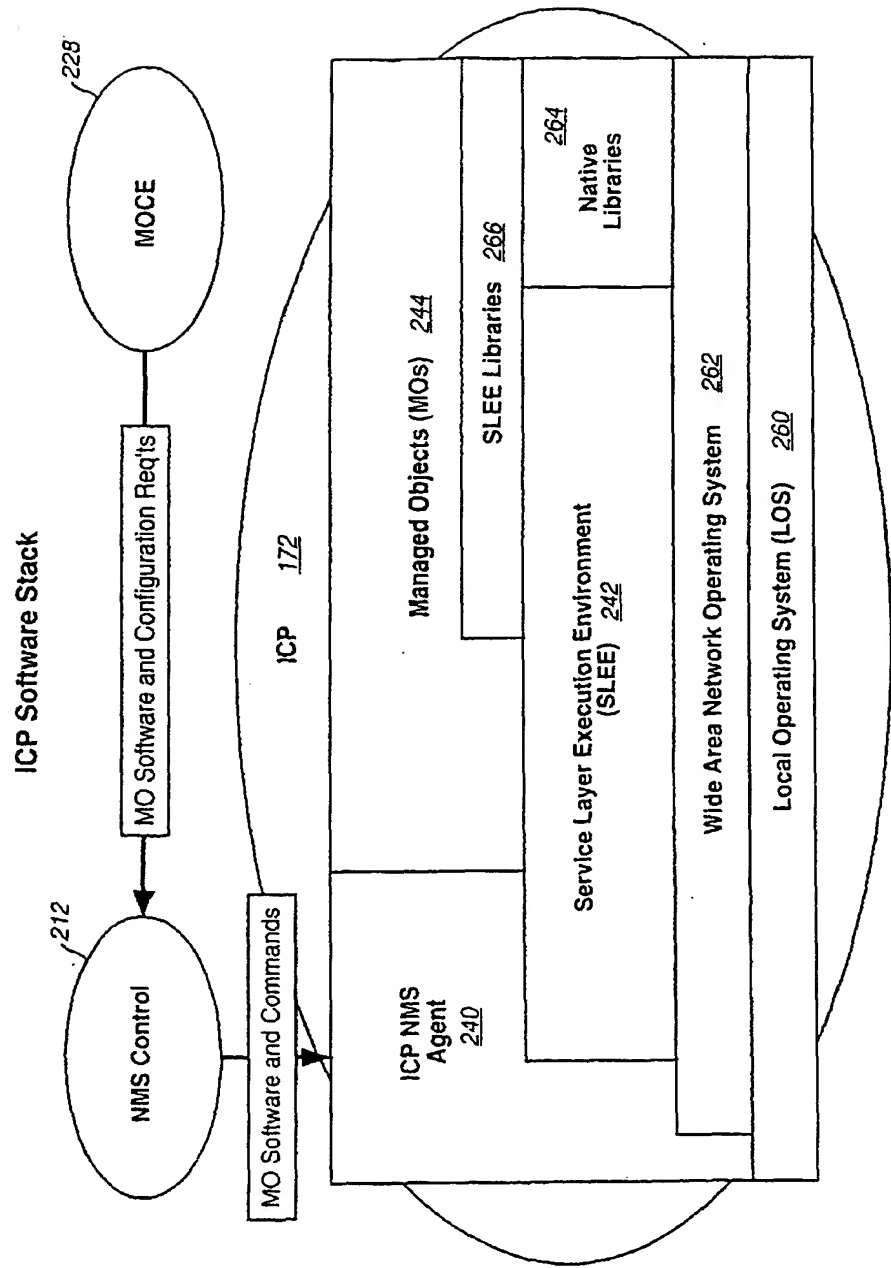


FIG. 6

(87)

【図 7】

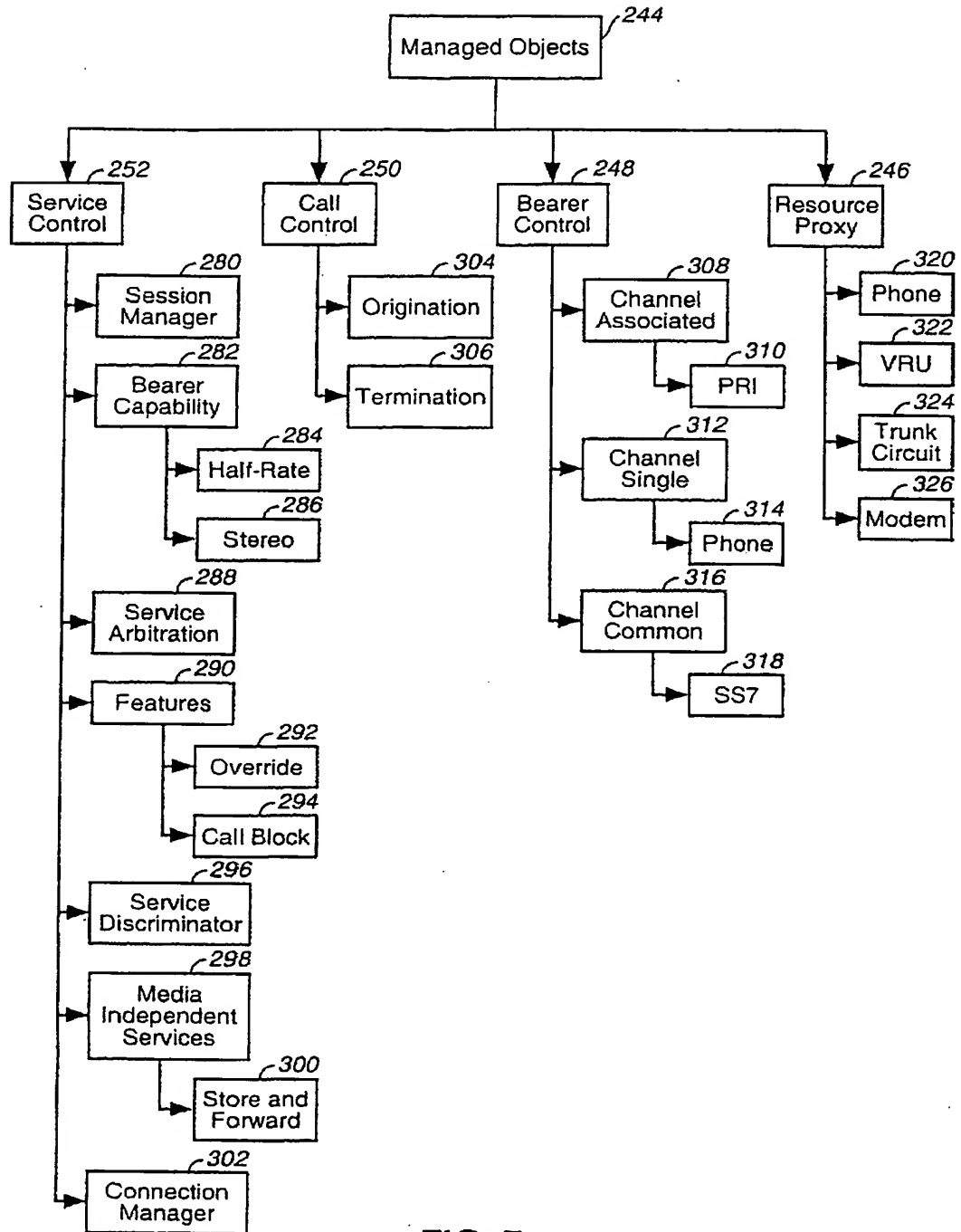
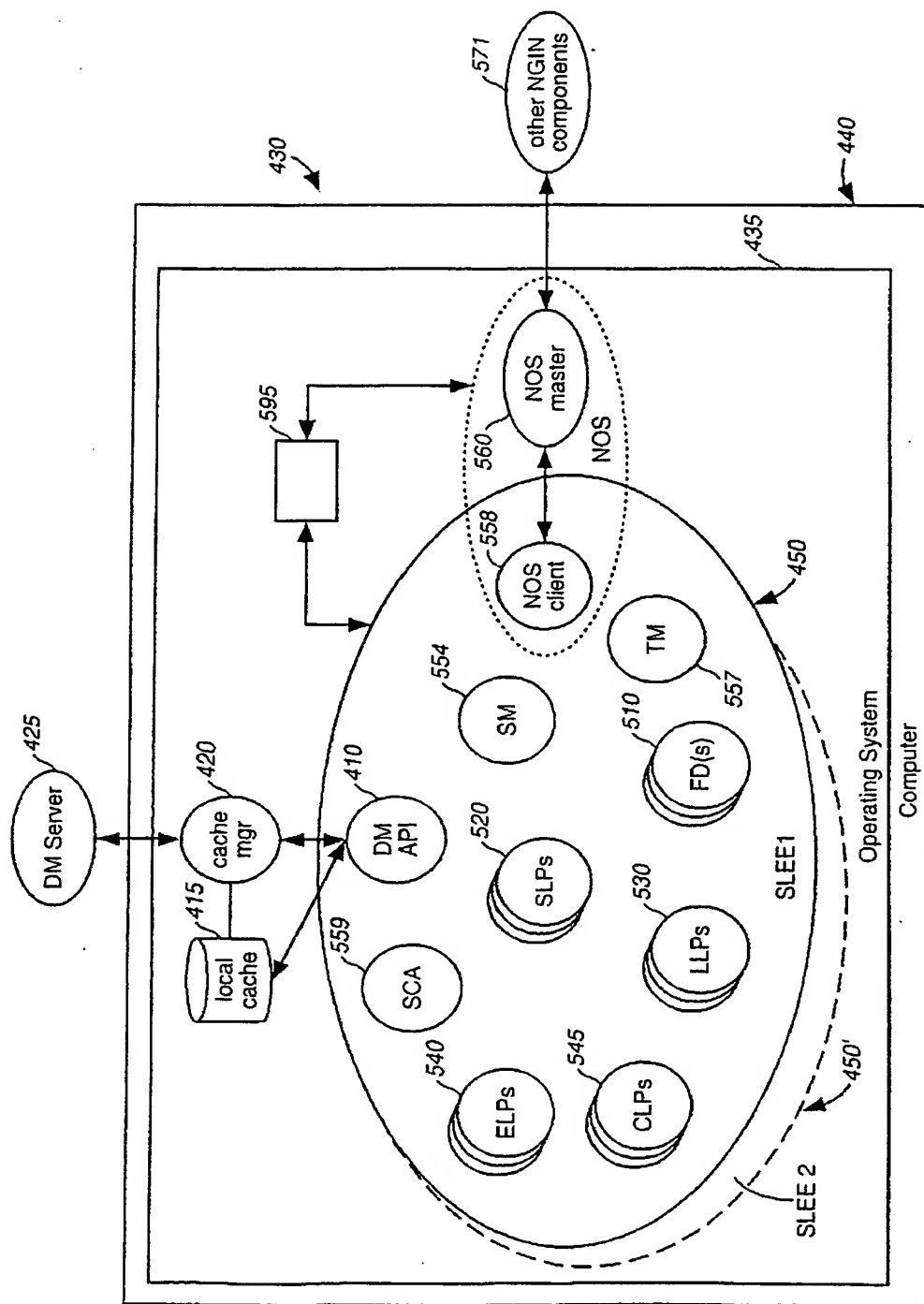


FIG. 7

(88)

【図 8】



**FIG. 8**

(89)

【図9】

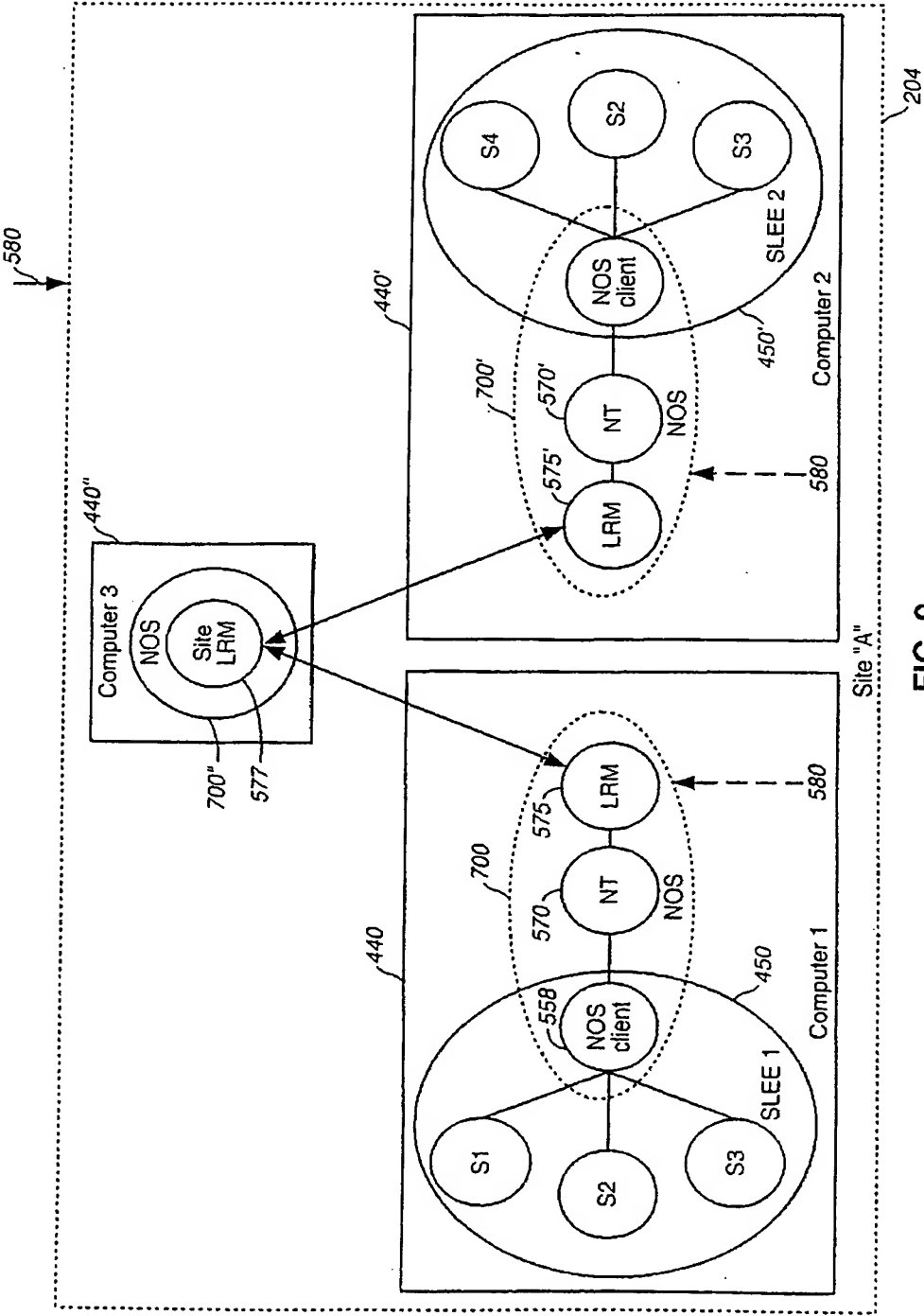


FIG. 9

(90)

【図 10】

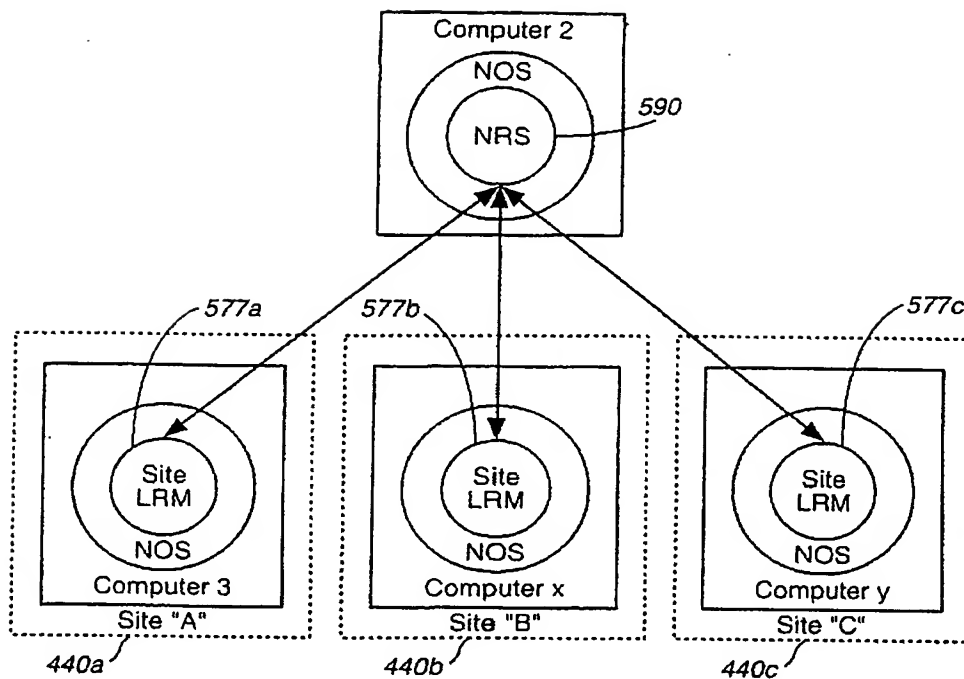


FIG. 10

【図 11a】

(91)

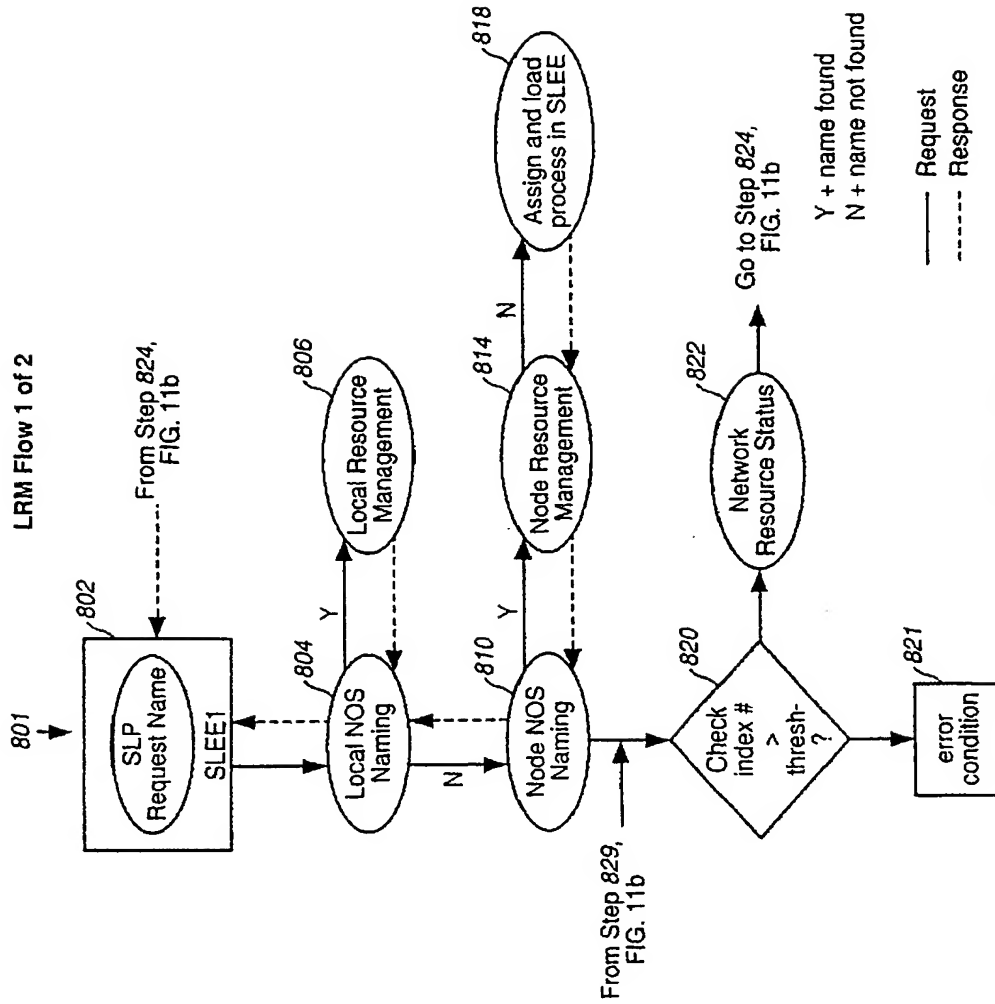


FIG. 11a

(92)

【図 11b】

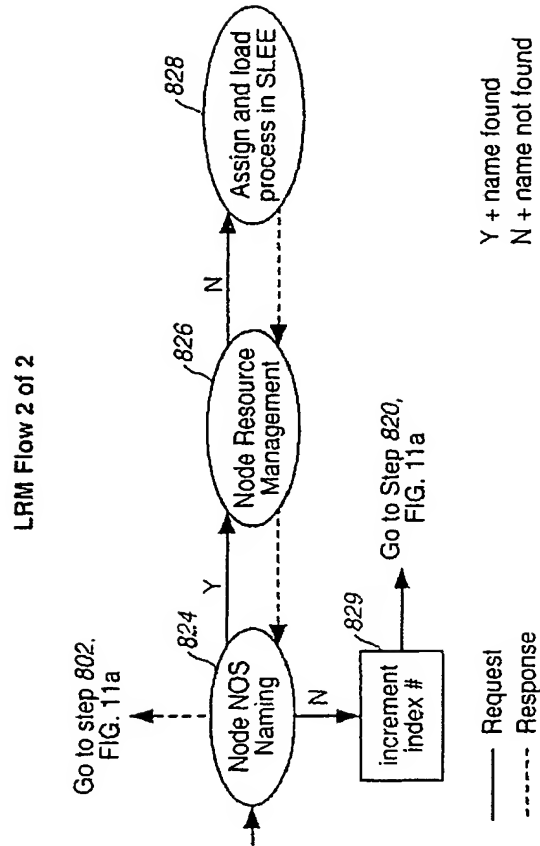


FIG. 11b



(93)

【図12a】

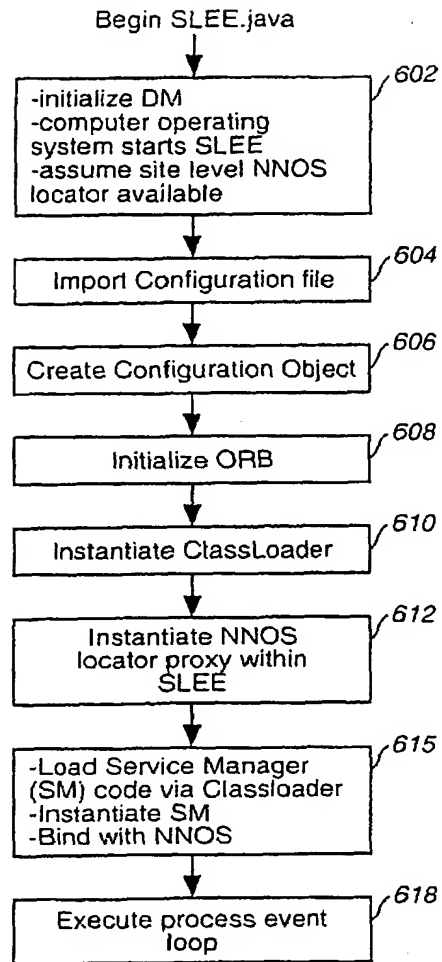


FIG. 12a

(94)

【図 12 b】

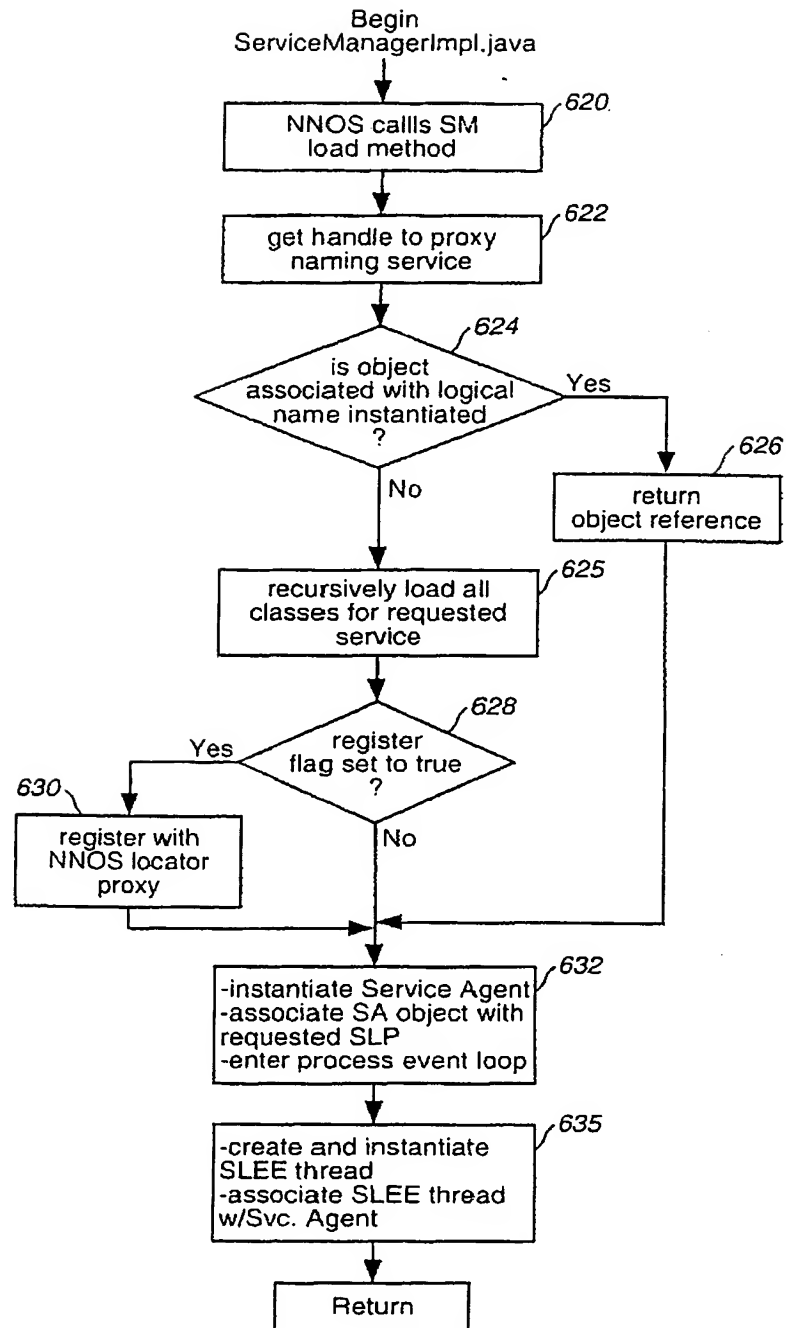


FIG. 12b

(95)

【図 12 c】

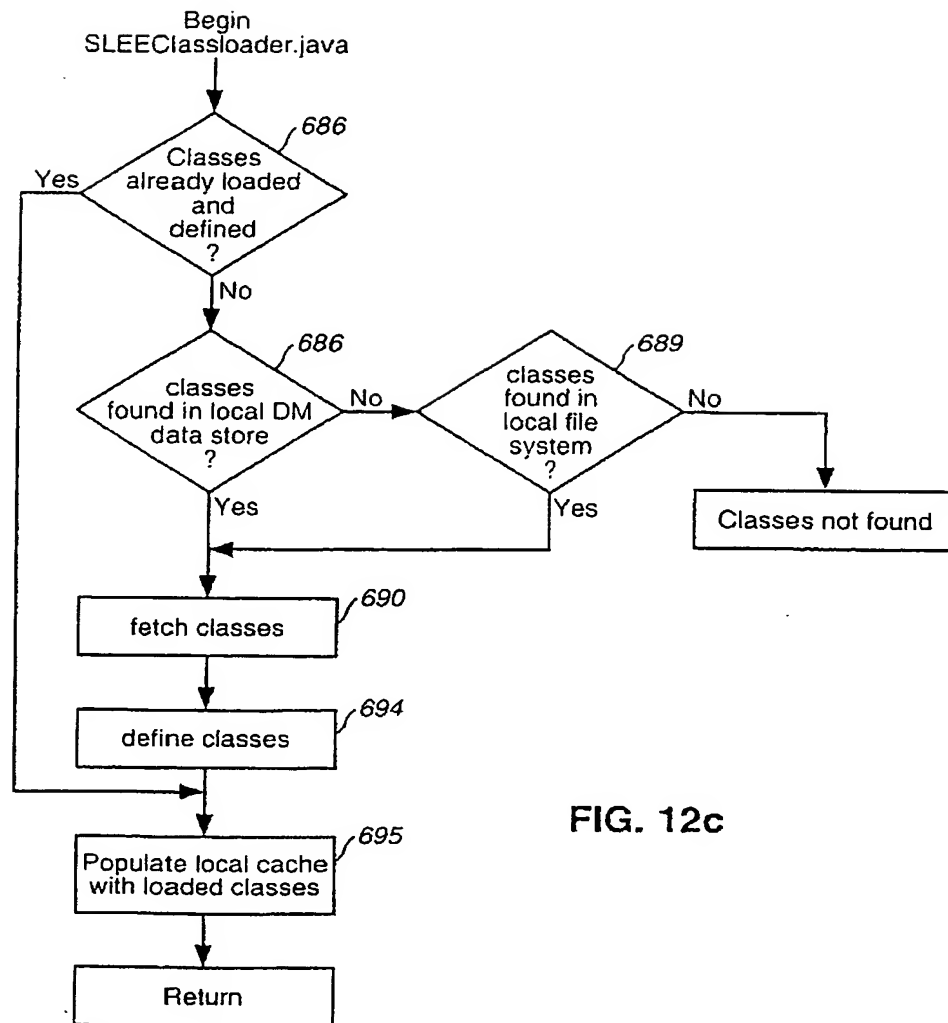
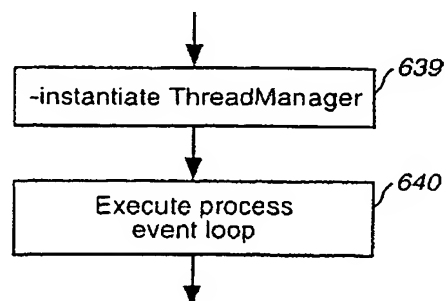


FIG. 12c

【図 12 d】

FIG. 12d



(96)

【図 12 e】

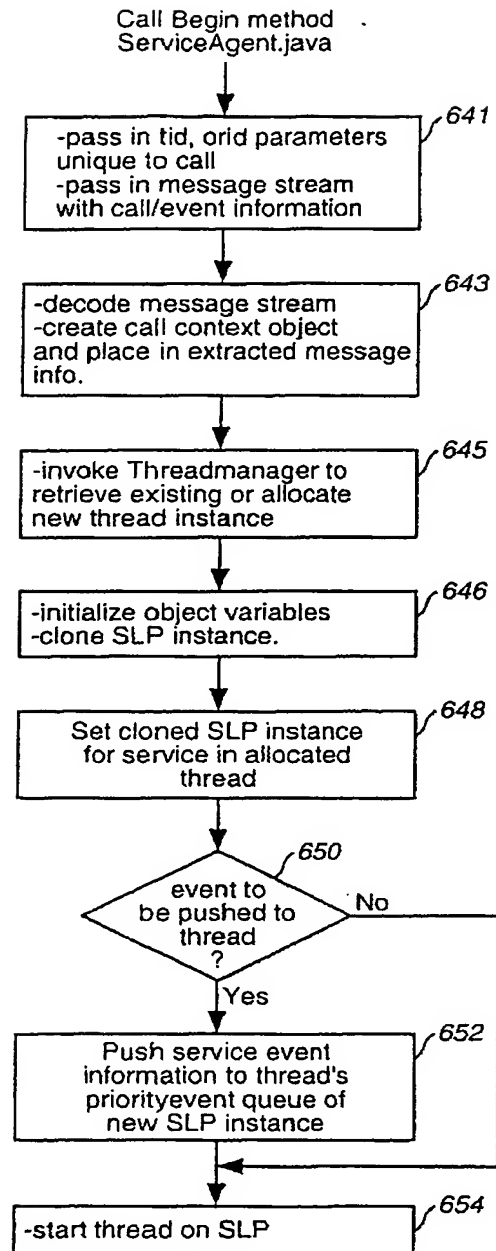


FIG. 12e

(97)

【図12f】

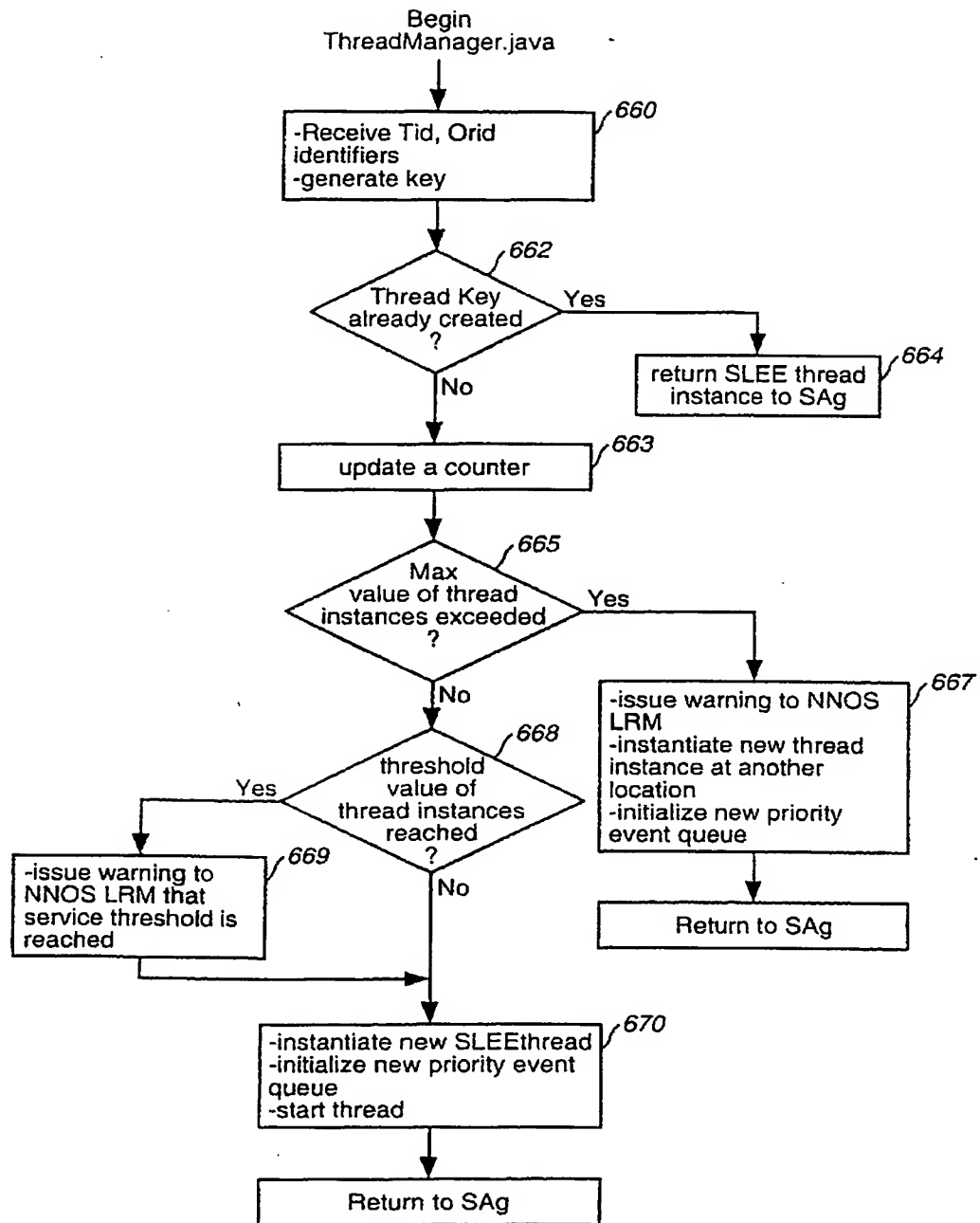


FIG. 12f

(98)

【図12g】

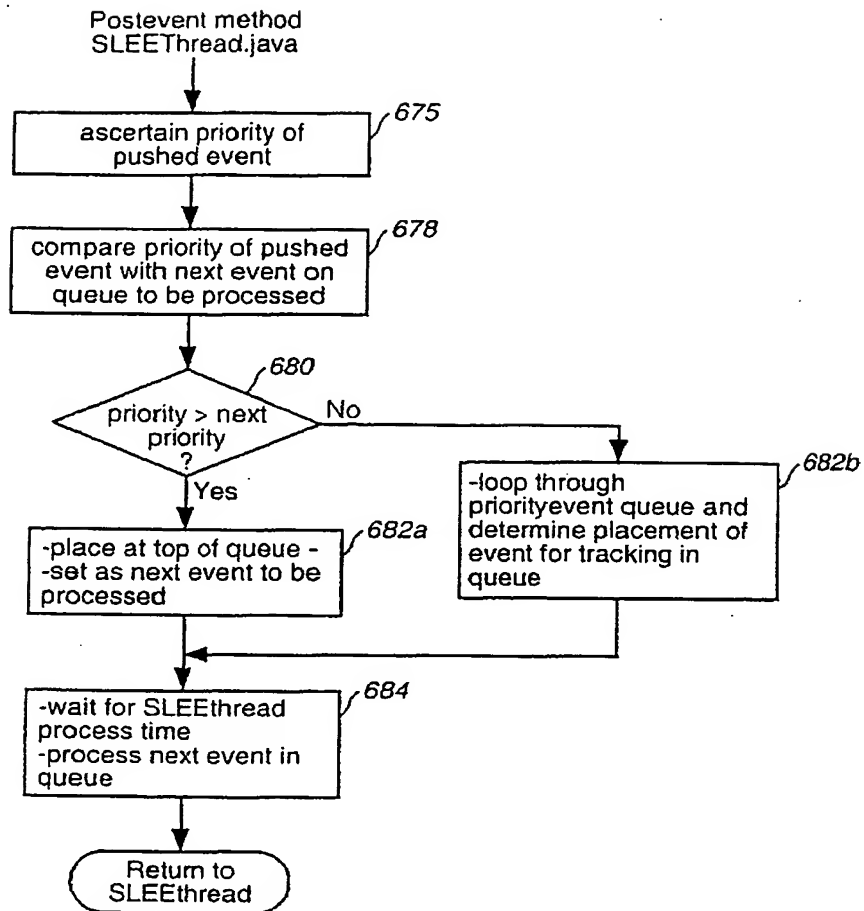


FIG. 12g

(99)

【図13a】

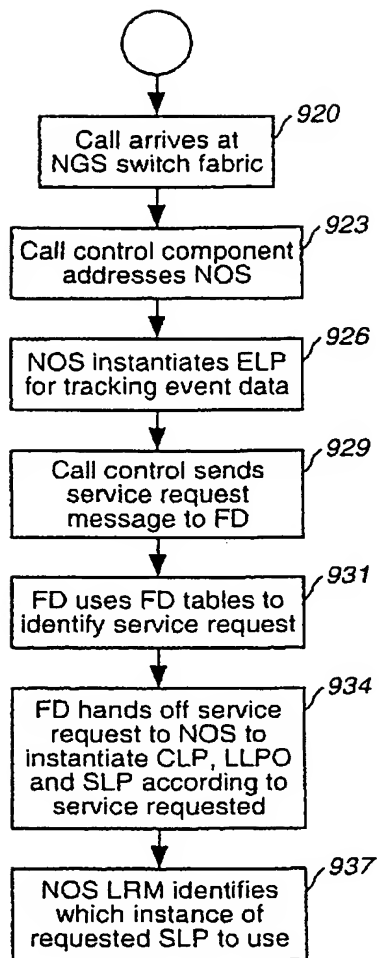


FIG. 13a

(100)

【図13b】

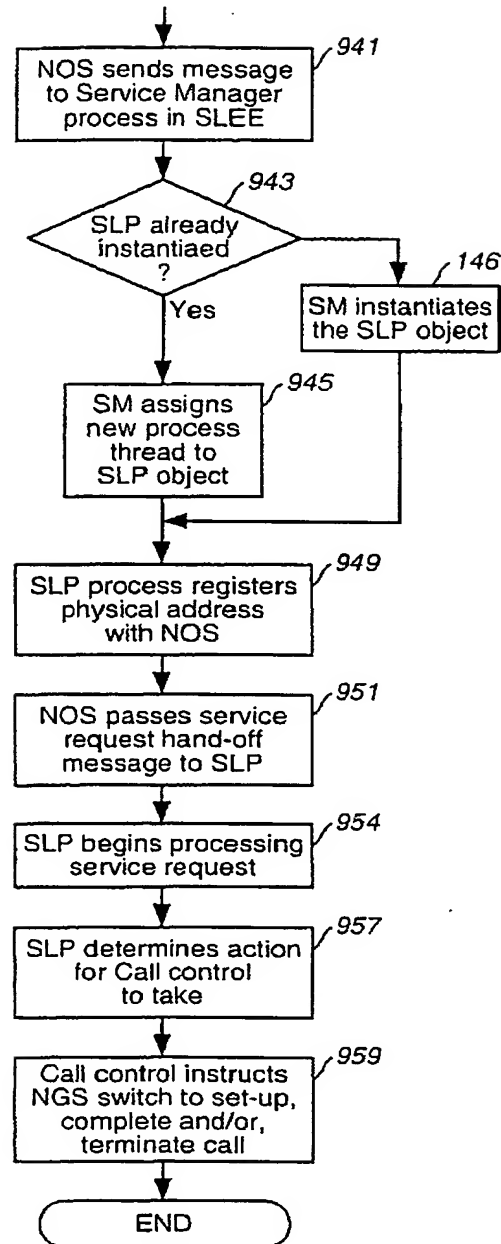


FIG. 13b



(101)

【図 13c】

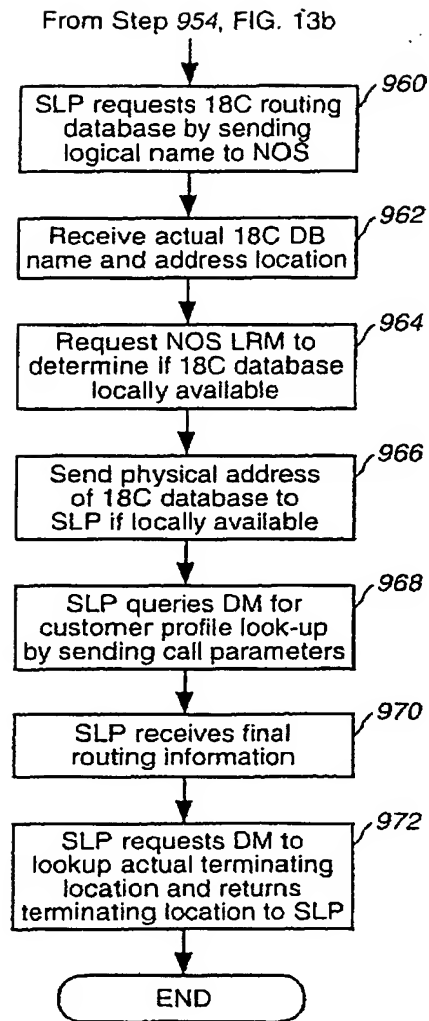


FIG. 13c

(102)

【図14】

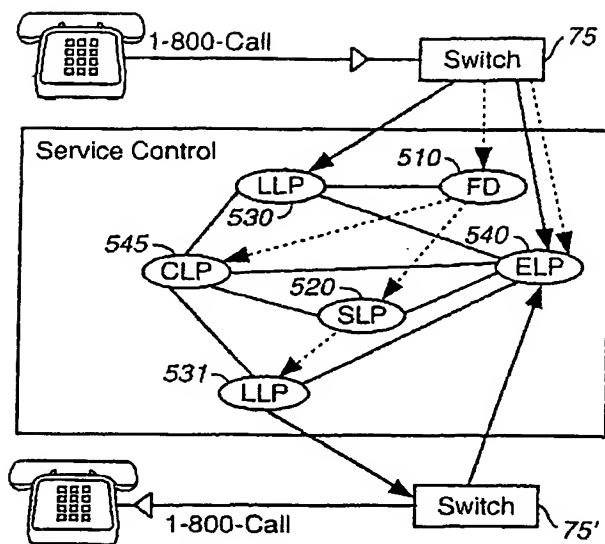


FIG. 14

(103)

【国際調査報告】

## INTERNATIONAL SEARCH REPORT

International application No.  
PCT/US99/24586

## A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) : G06F 13/00, 17/30; H04M 15/00, 3/54, 3/42; H04H 1/00;  
 US CL : 709/202, 217; 379/88.16, 112, 142, 211; 455/4.2; 707/2, 102  
 According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbol(s))

U.S. : 709/202, 217; 379/88.16, 112, 142, 211; 455/4.2; 707/2, 102

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y,E	US 5,999,965 A (KELLY) 07 December 1999, col.4, lines 1-45, col.6-21.	1-16
Y	US 5,335,268 A (KELLY, JR, et al.) 02 August 1994, col.2, lines 1-30, col.3-5.	1-16
A,P	US 5,825,865 A (OBERLANDER et al.) 20 October 1998, col.2-10.	1-16
A	US 5,619,557 A (VAN BERKUM) 08 April 1997, col.2-4.	1-16
A	US 5,754,939 A (HERZ et al.) 19 May 1998, col.4-8.	1-16
A	US 5,644,629 A (CHOW) 01 July 1997, col.3-10.	1-16

☐ Further documents are listed in the continuation of Box C.☐ See patent family annex.

\* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"B" earlier document published on or after the international filing date

"L" document which may throw doubt on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T"

later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X"

document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y"

document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"Z"

documents member of the same patent family

Date of the actual completion of the international search

13 JANUARY 2000

Date of mailing of the international search report

04 FEB 2000

Name and mailing address of the ISA/US  
 Commissioner of Patents and Trademarks  
 Box PCT  
 Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer

GLEN BURGESS

Telephone No. (703) 305-4792

Form PCT/ISA/210 (second sheet) (July 1992) \*

(104)

## フロントページの続き

(81) 指定国 EP (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OA (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG), AP (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), EA (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW

(71) 出願人 サミ・サイド

アメリカ合衆国・コロラド・80919・コロラド・スプリングス・ラグリー・コート・125

(72) 発明者 アジェイ・デオ

アメリカ合衆国・テキサス・75056・ルイスヴィル・サー・トリスタン・レーン・2508

(72) 発明者 ウェンディ・ウォン

アメリカ合衆国・テキサス・78287・ダラス・ケイブ・コーラル・ドライブ・4816

(72) 発明者 ヘンリー・ワン

アメリカ合衆国・コロラド・80919・コロラド・スプリングス・ガーデン・オブ・ザ・ガッツ・ロード・2424

(72) 発明者 サミ・サイド

アメリカ合衆国・コロラド・80919・コロラド・スプリングス・ラグリー・コート・125

Fターム(参考) 5K024 AA01 AA71

5K026 CC07 FF02 FF03 GG01

5K030 GA08 GA15 HA08 HB17 HC02

HD05 HD06

5K051 BB01 BB02 CC01 CC02 DD01

EE01 EE02 FF01 HH04

Subject: R : FE02-02090US01原稿チェック依頼送付のお知らせ。

Dat : Wed, 10 Sep 2003 08:33:02 +0900

From: Tsutomu Ishii <tsutomu.ishii@fujixerox.co.jp>

To: 中村 麻衣子 <Nakamura.Maiko@fujixerox.co.jp>

CC: 野見山 孝 <takashi.nomiyama@fujixerox.co.jp>, 吉沢 淳子 <Junko.Yoshizawa@fujixerox.co.jp>, Tsutomu Ishii <tsutomu.ishii@fujixerox.co.jp>

知財部  
中村様、野見山様

案件番号FE02-02090US01外国出願案件について、英文原稿チェック依頼を本日まで納期で頂いておりましたが、フェアやトラブル対応に追われチェックが終了していません。  
申し訳ありませんが、今週12日まで納期延長をお願いいたします。

以上。

On Wed, 03 Sep 2003 17:13:56 +0900

中村 麻衣子 <Nakamura.Maiko@fujixerox.co.jp> wrote:

> 研究本部 先端デバイス研究所  
> 石井様  
>  
> お疲れ様です。  
> 石井様が筆頭発明者となっている。  
> 案件番号FE02-02090US01外国出願案件について  
> 英文原稿チェック依頼を、社内メールにて送付致しました。  
>  
> チェックをして頂いて、下記期限までに  
> 中井 知財部 野見山 (7-992-2271) まで送付願います。  
>  
> 発明者処理期限 2003年 9月 9日  
>  
> お忙しいところ申し訳ありませんが、よろしくお願い致します。  
>  
> W W W W W W W W W W W W W W W  
> 中村 麻衣子  
> 富士ゼロックス(株) 中井事業所 知的財産部  
> TEL: 0465-81-8959 (内線) 7-992-8959  
> FAX: 0465-80-2491 (内線) 7-992-2491  
> e-mail Nakamura.Maiko@fujixerox.co.jp  
> W W W W W W W W W W W W W W W  
>  
>

石井 努 <Tsutomu Ishii>  
富士ゼロックス株式会社  
研究本部先端デバイス研究所  
TEL 046-238-3112 内線 519833  
FAX 046-238-4647  
E-mail tsutomu.ishii@fujixerox.co.jp

1. JP,2002-528932A

## NOTICES \*

Japan Patent Office is not responsible for any damages caused by the use of this translation.

. This document has been translated by computer. So the translation may not reflect the original precisely.

. \*\*\*\* shows the word which can not be translated.

. In the drawings, any words are not translated.

---

LAIMS

---

## Claim(s)]

Claim 1] Respectively, it is memory storage. The execution environment for giving one's service by answering exception of the event in the network switching device relevant to the service node. It is the system equipped with the above. The service profile containing the type and amount of service object resources relevant to the service processing in each node is generated for every node. The service administrator for downloading the aforementioned type and amount of service object resources to the aforementioned node based on the aforementioned profile, The instance generation mechanism for preparing for the execution in the aforementioned one or more execution environments, and performing instance generation of a service object, The execution-environment resources in service node are pursued, and the list of service types which can be used is maintained in each service node of the aforementioned network. According to a related functional situation service type [ each ] The resource-management equipment for indicating whether the service demanded can use for the instance generation in a service node is contained. According to the aforementioned functional situation When it indicates that the service demanded cannot use for the instance generation in the aforementioned network, the aforementioned resource-management equipment It tells that a new service object needs to be instance generated to the aforementioned central administrator, and is characterized by making the download and starting of new service in a service node carry out.

Claim 2] In the aforementioned instance generation mechanism One or more service objects are loaded from the aforementioned memory storage system. The 1st object for preparing for the execution in the aforementioned execution environment, and performing instance generation of the one or more aforementioned objects, Correspond to specific service and one or more service threads are assigned to each service instance corresponding to each request to receipt about the service. The system according to claim 1 characterized by containing the 2nd object by which the identifier of uniqueness is matched with each service thread instance.

Claim 3] Furthermore, the thing included for the Network Operating System for performing real-time transfer of a message and an event between the object instances under execution, The 2nd object of the above corresponding to specific service performs the event between the aforementioned service instances, and the channeling of a message, the system according to claim 2 characterized by containing the aforementioned uniqueness identifier for matching a receiving message and an event with a proper service instance in the aforementioned event and a message.

Claim 4] Furthermore, the event queue mechanism for putting into a queue the event matched with the aforementioned service instance which was assigned for every service thread instance, and which receives during service execution is included. The system according to claim 3 carry out that there is related priority which directs the turn that the aforementioned event should be carried out in an event, and that the aforementioned event queue equipment makes processing of a receiving event possible based on the related priority as the feature.

Claim 5] Furthermore, the system according to claim 3 characterized by that the class loader process which loads one or more service objects from the aforementioned memory storage system is first included according to the composition rule which carries out the initial service function about the aforementioned service node, and for the aforementioned class loader to undertake the duty perform instance generation of the 1st object of the above which can be used, and arbitrary service objects, according to a prior definition service function strategy.

Claim 6] The system according to claim 3 characterized by containing the Thread Manager instance for generating the alarm signal to the aforementioned resource-management equipment when the amount of the thread instance relevant to a certain service is compared with the predetermined threshold calculated in the aforementioned service profile and instance generation of a new service thread instance is not supported any longer in the aforementioned execution environment by the 2nd object of the above corresponding to specific service.

Claim 7] The system according to claim 6 characterized by that the aforementioned Network Operating System is contained in the aforementioned service object instance generation mechanism, and for the aforementioned resource-

management equipment pursuing the throughput of the execution environment in each service node further, and displaying whether execution of a certain service is possible on the aforementioned Network Operating System by the execution environment in a certain service node based on the throughput.

Claim 8] The system according to claim 7 by which the aforementioned resource-management equipment is further characterized by transmitting overload situation display to the aforementioned Network Operating System, and preventing instance generation of the service object beyond it in the aforementioned execution environment if a threshold predetermined [ aforementioned ] in the number of service threads under present execution is exceeded in a certain execution environment.

Claim 9] In the aforementioned instance generation mechanism The registry of the active service object thread corresponding to the instance of the service performed by the execution environment prepared in each aforementioned execution environment, The wafer scanner for mapping the service logical name equipped with refer to the object is contained, The system according to claim 3 characterized by the aforementioned Network Operating System enabling instance generation of the service object thread instance demanded in a partial execution environment using aforementioned refer to the object.

Claim 10] Reception of the event in memory storage and the network switching device relevant to the service node is answered for every service node. It is the method for offering service in the service node of a communication network at the execution environment for giving one's service was prepared. The service profile containing the type and amount of service object resources relevant to the service processing in each node is generated for each service node of every. The step which downloads the aforementioned type and amount of service object resources to the aforementioned node based on the aforementioned profile, Step which prepares for the execution in the aforementioned one or more execution environments, and performs instance generation of a service object By maintaining the list of service types which can be used in each service node The execution-environment resources in a service node are pursued. according to a related functional situation service type [ each ] The step as which it is displayed whether the service demanded can use for the instance generation in a service node is contained. According to the aforementioned functional situation If it indicates that the service demanded cannot use for the instance generation in the aforementioned network The method characterized by telling that a new service object needs to be instance generated to central administrator equipment, and making the download and starting of a new service object in a service node carry out.

Claim 11] The method according to claim 10 characterized by containing the step characterized by providing the following. To the aforementioned instance generation step Step which prepares the 1st object for loading one or more service objects from the aforementioned memory storage system, preparing for the execution in the aforementioned execution environment according to the service request which received, and performing instance generation of the one or more aforementioned objects. The 2nd object which corresponds to specific service and assigns one or more service threads to each service instance corresponding to each request to receipt about the service and by which the identifier uniqueness is matched with each service thread instance.

Claim 12] Furthermore, the method according to claim 11 characterized by that support service processing and the step which transmits the message and event which are generated during execution of a service object between the service objects under one or more execution is contained, and for the aforementioned event and a message to be discriminated by the aforementioned uniqueness identifier, and to amend the service instance under execution through the 2nd object of the above.

Claim 13] Furthermore, the method according to claim 12 characterized by that the step which is received during service execution, and which puts the event matched with the service instance under execution into a queue is contained, that there is related priority which directs the turn that the aforementioned event should be carried out in the aforementioned event, and to process the aforementioned receiving event based on the corresponding priority.

Claim 14] Furthermore, the method according to claim 10 of carrying out that the step which loads one or more service objects from the aforementioned memory storage system is first contained according to the composition file which offers the initial service function about the aforementioned service node, and that the aforementioned class loader undertakes the duty in\_ which to the aforementioned class loader performs instance generation of the 1st object of the above which can be used, and arbitrary service objects according to a prior definition service function strategy as the feature.

Claim 15] To the aforementioned step which pursues the execution-environment resources in a service node Step which compares the amount of the thread instance relevant to a certain service with the predetermined threshold calculated in the aforementioned service profile, Method according to claim 11 characterized by to contain the step which generates the alarm signal to resource-management equipment in the aforementioned execution environment when instance generation of a new service thread instance is not supported any longer.



[Claim 16] In the aforementioned instance generation mechanism The step which maintains the registry of the active service object thread corresponding to the instance of the service performed by the execution environment prepared in each aforementioned execution environment, The step which maps the service logical name equipped with refer to the object, The method according to claim 10 characterized by containing the step which enables instance generation of the service object thread instance demanded in a partial execution environment using refer to the aforementioned object.

---

[translation done.]

## NOTICES \*

Japan Patent Office is not responsible for any damages caused by the use of this translation.

This document has been translated by computer. So the translation may not reflect the original precisely.

\*\*\*\* shows the word which can not be translated.

In the drawings, any words are not translated.

## DETAILED DESCRIPTION

## Detailed Description of the Invention]

0001] [The technical field to which invention belongs]

This invention relates to the new service control system which generally offers real-time event processing service in each of two or more service nodes specifically distributed to the whole intelligent network about the intelligent network system for offering communication service.

0002] [Description of the Prior Art]

Network services are functions performed by communication networks, such as data and a telephone, and the resource relevant to it according to a dialog with one or more subscribers. For example, when a subscriber dials the number of special turn, telephone network permanent residence service of an automatic transfer, voice mail access, etc. can be started. Other network services can be turned to a network owner's support by security, validity check, and authentication. An addition and correction of service need change of a communication network.

0003] The customary communication network consists of the switches and communication service by which interconnection was carried out. Such a switch is controlled by the united processor which operates with the original software or the original firmware which the switch maker designed, or the embedded processor. Generally, a switch maker's software or firmware must be supporting all the functional sides of service processing, call processing, facility processing, and network administration. That is, when a network owner wants to realize new service or to correct the existing service, various switch makers have to revise the software of all the switches in a network.

0004] Since a network contains various switch models by various makers, it is necessary to develop new software carefully, to examine and to develop. Since the code size of each switch is becoming complicated greatly for every new revision, time required to develop and examine new software and develop is becoming complicated for a long time. Therefore, there is this several years thing in this process. Furthermore, a burden may start the processor of a switch, the possibility of the malfunction of a switch may increase, and repair and exchange of a switch may be needed with this high complexity. Furthermore, in order that two or more network owners may be dependent on the switch maker of common group, two situations which are not desirable of restricting competition arise.

0005] A maker's software release may incorporate [ 1st ] change demanded by some network owners, therefore a network owner cannot distinguish his service and the service offered from those competition truly. Moreover, thereby, some network owners are kept waiting until a maker includes the demand by other network owners in a new release. The software release of a switch which included the function according to the demand from a certain network owner in the 2nd, and realized new service has a bird clapper to others and a network owner possible [ use ], without meaning.

0006] Since the need of new network services has increased exponentially in [ past ] five - ten years, it is impossible to permit such a problem by a rise of a subscriber's volatility, the kind of traffic and increase of bandwidth, decay of a traditional numbering method, the advancement of service, and rise of competition. Therefore, it is widely recognized that it is necessary to take in a flexible method rather than new network architecture creates service logic, develops and reforms. In order to fully understand the new architecture of this invention explained later, the following explanation of the related conventional technology is offered with reference to drawing 1 .

0007] With reference to drawing 1 , the logical expression of various switching architecture containing this invention is shown. The integral switch in which the whole was shown by 20 contains the service processing facility 22, the call processing function 24, the facility processing facility 26, and a switch mechanism 28. Altogether, as symbolized into the group 30, the hard code was carried out and it was mixed, and these functions 22, 24, 26, and 28 have turned drawing 1. Furthermore, functions 22, 24, 26, and 28 are designed by the switch maker, and operate on the original platform which changes with makers. Consequently, these functions 22, 24, 26, and 28 are uncorrectable without a maker's support, and this delays development and mounting of service and is raising the cost which takes out new service to a commercial scene. Therefore, development of innovative new service, call processing, data processing, signal

rocessing, and network operation receives the limit by management of the maker to a maker's original switch hardware and original software, and the original difficulty which establishes an industry standard and is realized.

[0008] The service processing facility 22 is coded within the integral switch 20, and makes possible only the local control of the local contents of data, and the process based on the dialed number. This local information is interpreted with the process engine which performs the coded service function and by which the hard code was carried out. The hard code of the call-processing function 24 is carried out, and it offers a call origination function and a call-in function. This process establishes each connection and makes a call actually complete. Similarly, the hard code also of the facility processing facility 26 is carried out, and all data processing relevant to the physical resource contained in a call is realized. A switch mechanism 28 expresses the hardware component of the switch which performs integral software offered by switch makers, such as Northern Telecom, and a computer. A switch mechanism 28 offers a physical mechanism required in order to establish connection, and although it contains a transport unit (TI and DSO), switch INGU matrix equipment (a network flat surface and its processor), a link layer signal processor (SS7, MTP, SD, LAPD), and a personal circuit (a conference port, voice tone detector), it is not limited to these. [0009] In the attempt which tackles the above-mentioned technical problem, International Telecommunication Union and European Telecommunication Standards Institute recognized ITU-T intelligent network specification (Intelligent Network Standard) ("IN"). Similarly, Bellcore recognized advanced intelligent network specification (Advanced Intelligent Network Standard) ("AIN"). Such two specification has the almost same purpose and fundamental concept, although the states of expression and development differ. Therefore, it is considered that such specification is one network architecture into which the service processing facility 22 was separated from the switch.

[0010] Probably by using IN and AIN architecture, it is possible to exhibit new service by creating the new service logic program (Service Logic Program) ("SLP") which is the table of a composition unit (Service Independent Building Blocks) ("satellite infrared spectrometer") independent of the service essentially started in a given type of call, and developing. By this technique, some specific element types carry out mutual operation with SLP, and provide a network subscriber with service. Consequently, the service which is possible or it is new is restricted by the existing IBBS.

[0011] IN or AIN architecture the whole was indicated to be by 40 divides the function of the integral switch 20 into the service control point (Service Control Point) ("SCP") 42, the service switch INGU point (Service Switching Point) ("SSP"), and the switching system 44 logically. Although SCP42 contains the service processing facility 22, SSP and the switching system 44 contain the call-processing function 24, the facility processing facility 26, and a switch mechanism 28. In this case, as symbolized by the group 46, the hard code was carried out and it was mixed, and the call-processing function 24, the facility processing facility 26, and the switch mechanism 28 have turned drawing 1.

[0012] The service switching point ("SSP") is a functional module in a switch, in order that a subscriber's signal dispatch may recognize needing routing which is not a simple thing only based on the dialed number. SSP interrupts the further operation of a call, performs the inquiry about suitable processing of a call to remote [ 42 ] one SCP simultaneously, and is essentially \*\*\*\*\* as a database server of some switches. The load of the task which time requires although it is rare which is for processing a special service call is mitigated from a switch by division of this processing. Furthermore, the balance with the thing which work for the whole network and for which it can correct easily and has one repository with a heavy burden, and developing the perfect copy of the repository in all switches can be taken by this moderate centralization.

[0013] Next, when drawing 2 is referred to, the communication system which uses IN or AIN architecture is shown, and the whole is expressed with 50. Various customer systems, such as the ISDN terminal 52, the 1st telephone 54, and the 2nd telephone 56, are connected to SSP and the switching system 44. The ISDN terminal 52 is connected to SSP and the switching system 44 by a signal line 60 and the transmission line 62. The 1st telephone 54 is connected to SSP and the switching system 44 by the transmission line 64. The 2nd telephone 56 is connected to the remote switching system 66 by the transmission line 68. Moreover, the remote switching system 66 is connected to SSP and the switching system 44 by the transmission line 70.

[0014] As drawing 1 was explained above, SSP70 is a functional module in a switch, in order that a subscriber's signal dispatch may recognize needing routing which is not a simple thing only based on the dialed number. SSP70 interrupts the further processing of a call and performs the inquiry (query) for suitable processing of a call. This inquiry is sent to remote [ 42 ] one SCP in the form of SS7 message. By changing the contents of a data base in the place, since the service control point 42 can change the network function which the subscriber connected through the switch with which many are inherent can use, it has a name such. Through a signal line 72, an inquiry is sent to the signal transfer point (Signal Transfer Point) ("STP") 74 which is only a router for SS7 messages in such an element, and, next, is sent to SCP42 through a signal line 76.

[0015] It conceives of the integrated service managerial system (Integrated Service Management System) ("ISMS") 78

is a management tool which develops or changes service or manages access to the service for every subscriber. SMS78 operates by mainly changing the operation logic and data which were memorized by SSP70 and SCP42. SMS78 has various user interfaces 80 and 82. It connects with SCP42 by the operating line 84, and connects with SSP and the switching system 44 by the operating line 86, and this ISMS78 is connected to the intelligent peripheral device (Intelligent Peripheral) ("IP") 88 by the operating line 90. The intelligent peripheral devices 88 are equipments used in order to add the function which cannot be used on a switch to a network, such as a voice recognition and response system and a speech recognition system. IP88 is connected to SSP and the switching system 44 by a signal line 92 and the transmission line 94.

0016] Next, processing of the call by the conventional technology is explained with reference to drawing 2. A call is performed, when a customer takes an earphone and begins to dial. SSP70 in a corporate NOSUITCHI monitor supervises dialing, and recognizes a trigger sequence. SSP70 interrupts the further processing of a call until it can investigate service logic. Next, SSP70 constitutes SS7 standard message and sends it to SCP42 through STP74. SCP42 receives and decodes a message and starts SLP. SLI interprets SCP which can operate other functions, such as a database lookup for number conversion. SCP42 sends out a message to a network element, in order to return SS7 message about processing of the call to SSP and the switching system 44 or to perform suitable service. In order to cut a call, SS7 message is sent between switches in the end of a call, and each switch which participates in a call creates the detail record of a call. For every call, it is collected, and is related and solved off-line, accounting of a long distance call is drawn by that cause, and processing of a call completes the detail record of a call.

0017] IN and AIN architecture tend to define 1 set of standard functions in advance, in order to support all services that can be predicted. The hard code of such a standard function is altogether carried out to various state machines in a switch. though regrettable, the new function which may be produced in relation to new technology or an unexpected service request is unrealizable without an examination in the wide range overhaul of the network software crossed to many vender plat forms Furthermore, when a new function needs standardized change of as opposed to [ call and ] a model, a protocol, or an interface, realization of service using the function may be overdue until the change is recognized by the industry standard group. However, although the draft standard is going to expand the functional set which IN and AIN support, the supplier company of equipment has refused supporting such a draft standard for wonderful increase of the complexity of a code.

0018] If drawing 2 is referred to further, other limits of IN and AIN architecture will be produced by having, the call-processing function and the facility processing facility 70, i.e., SSP, which operates within a switch. Consequently, these functions must be offered by each switch maker using original software. Therefore, the network owner is still greatly dependent on a maker software release, in order to support a new function. It complicates a problem further that a network owner cannot examine SSP70 module together with other modules in the development and the test atmosphere which are integrated. Furthermore, there is no guarantee of 70 SSPs for a switch maker's processing environment to which a network owner's service creation environment and compatibility are said more.

0019] Thus, when two or more network owners are dependent on the switch maker of a common group, two situations which are not desirable of restricting competition arise. A maker's software release may incorporate [ 1st ] change demanded by some network owners, and, thereby, a network owner cannot distinguish from the service provided with its service by competition truly. Some network owners are kept waiting until a maker includes the demand from other network owners in a new release by this. The switch software release including the function in which mounting of new service was required from a certain network owner the 2nd has a bird clapper to others and a network owner possible use ], without meaning. Therefore, since a network owner can control completely the functional element which forms the behavior of network services or cannot use in spite of the intention of IN and AIN architecture, creation of new service of a network owner, an examination, and expansion are still barred.

0020] In one attempt which solves these problems, the individual switch intelligence and switch mechanism (Separate Switch Intelligence and Switch Fabric) ("SSI/SF") the whole was indicated to be as 150 ( drawing 1 ) separate SSP70 from the switching system 44 logically. Next, when drawing 1 is referred to again, the switch intelligence 152 is coded as an individual state table with the corresponding state-machine engine by which the hard code was carried out, and contains the call-processing function 24 and the facility processing facility 26 which were symbolized with circles 154 and 156. The connection between the switch mechanism function 158 and the switch intelligence function 152 is extensible with a communication network, consequently a switch mechanism 158 and the switch intelligence 152 do not need to be arranged together physically, do not need to be performed within the same processor, or do not need to correspond to further 1 to 1. As the result, the switch intelligence 152 is [ no ] common to switches, is peculiar to service, and offers the adjustable interface of the easy function which is not peculiar to a maker again.

0021] The intelligent calculation complex (Intelligent Computing Complex) ("ICC") 160 contains the service processing facilities 22 and 30 which communicate with two or more switch intelligence elements 152. Since almost

If basic functions are moved out of the field of a code peculiar to a maker, this technique provides a network owner with the advantage of flexible service mounting. The further improvement is realizable by offering the environment integrated more for creation of service logic, development, an examination, and execution.

0022] As considered above, the present network switch is based on original hardware and original software by the integral. Although this thing has the cost of millions of dollars, such a facility has processing speed comparatively slow [a network switch], when it sees from a viewpoint of the calculation technology in which the present use is possible. For example, such a switch communicates mutually using data communication protocols, such as X.25 which generally supports the transmission speed of 9.6kb(s)/a second among various platform forms switching within the net, based on reduction instruction set computing ("RISC") which operates in a 60MHz range. This is very late as compared with the personal computer containing the processor which operates above 200MHz, and the high-end computer workstation which offers 150mb[ /s] FDDI and an ATM interface a second. Therefore, a network owner needs to be able to use a high-end workstation instead of original hardware.

0023] [Problem(s) to be Solved by the Invention]

This invention is aimed at the service control system which offers the real-time service which processes all of the event and service request which were received in resource complex physically related with each of two or more distributed service nodes of an intelligent communication network, such as a switch and a router.

0024] [Means for Solving the Problem]

Generally, in processing of a service request, the service control component of this invention can be directed to intelligent network resource complex, such as an ATM switch, the Internet gateway, an intelligent peripheral device, other switches, or a router resource, and contains an intelligence required in order to process a service request further. In detail, a service control component can acquire information (service or user data) required to have a dialog with other intelligent network components, and access other logic, or process a service logic instance by the incorporated intelligence. In real-time service processing, it connects with resource complex and a local data management system, and service control has a dialog, and holds logic and a throughput required to process the attempt of the service which an intelligent network offers. Service control is managed, updated and managed with the service manager and data control component of an intelligent network. An intelligent network is not dependent on the call switching platform or resource complex which receives a call, offers transparent intelligent call-processing service to it again, and it suits easily so that a call event may be processed. Therefore, the dependence to hardware peculiar to an expensive vendor, an operating system, and a switching platform is lost. A distributed intelligent network supports execution of the event processing service independent of a place further, it enables it to perform a modular software logic program substantially in architecture anywhere, and communication independent of the place between such distributed processes is realized, therefore a special service node becomes unnecessary.

0025] In a detail, this invention controls more one or more processes started when a service request is sent to a service control component with resource complex. Service control has a dialog with other components, and accesses demand data required to offer demand service. A process is completed when it stops completing the behavior sequence of demand service or a service user using service. All the resources in connection with providing a service request side with service instead of a demand are opened wide in the end of processing. Each service request starts one instance (thread) of service processing, and, thereby, randomness or a lot of parallel processing with few bottlenecks is realized.

0026] As for the instance of each service thread, it is desirable to maintain about the specific call instance to the suitable service thread queue for memorizing and performing according to the predetermined priority relevant to the event by the service control which offers the asynchronous channeling event which received the event queue of itself. A thread instance is interrupted, when service control offers the asynchronous channeling of the event to a switch / resource complex further, or offers the execution service logic program of it and others together and waits for a response.

0027] According to this invention, the roles with a main service control component An event or a demand is received and processed from the external resource of a switching platform or others. In order to process a carrier beam demand, discriminate and start a service logic program, and a Network Operating System (NOS) is minded from data control storage. A data base application program interface (API) is minded. Or direct, Require the data relevant to service or a subscriber, and the data relevant to service or a subscriber are updated in a data control component through NOS. The function which sends the event and message which attached priority to the logic program of resource complex or others, and controls a user's dialog is offered. The message set which includes user inputs, such as a dual tone multi URIKENSHI DTME number corresponding to PIN, the selected menu item, etc., for example is received from resource complex. It includes maintaining the situation and data of all participation objects which are contained in the same service processing instance, generating a billing record, and sending it to the billing record generation function of

. data control component.

0028] The various features of the freshness by which this invention is characterized are shown in the claim which appends and constitutes a part of this indication in detail. In order to understand better the specific purpose attained by the advantage on this invention and its operation, and its use, please refer to the drawing and explanation which illustrated and explained the desirable operation form of this invention.

0029] He can understand other advantages better by referring to the following explanation with an accompanying drawing the above of this invention. [0030] [Embodiments of the Invention]

It sets on these specifications and this invention is intelligent distributed-network architecture (Intelligent Distributed Network Architecture) ("IDNA") or a next-generation intelligent network. (Next Generation Intelligent Network) ("NGIN") It is one component of the comprehensive intelligent network called instead. NGIN architecture is designed so that intelligent call-processing service may be performed to the call arbitrary type received on resource complex or switching plat forms, such as a switch, a router, and IP end address, so that this specification may explain. Including the service node which plurality distributed, IDNA/NGIN offers the execution environment which offers a call-processing function required to process the call, when each node receives a call in the switch or resource complex associated as physically as the specific service node. NGIN is architecture with a very high scalability, and it is designed so that it may develop certainly and the service object which is carried out as an independent service logic program (Service Logic Program) ("SLP") and which can be performed, and the related data for performing event services, such as a 1-800 telephone call and facsimile transmission, can be maintained by the high method of cost efficiency in a service node. by using the object request broker technology of CORBA conformity, an intelligent network supports execution of the call-processing service which becomes independent of the event switching plat form or resource complex which receives an event or a call, and is not dependent on a transparent place and a transparent plat form to it, and performs the logic program of a high level as a matter of fact anywhere on a network nothing with regards to a service execution plat form -- things are made possible Furthermore, a system offers the communication which is not dependent on a place between such distributed processes.

0031] Next, generally reference of drawing 1 shows intelligent distributed-network architecture (Intelligent Distributed Network Architecture) ("IDNA") as 170. this invention unifies ICC160 and the switch intelligence 152 of the SSI/SF architecture 150 to the intelligent call processor ("ICP") 172. Unlike IN or AIN of the SSI/SF architecture 150 by which the function was defined on the state table, ICP172 contains the service control function 22, the call-processing function 24, and the facility processing facility 26 as a managed object in the object-oriented plat form symbolized by blocks 174, 176, and 178. ICP172 is logically separated from the resource complex 180.

0032] Next, with reference to drawing 3 , the communication system using the intelligent distributed-network architecture by this invention is shown, and the whole is shown as 200. A wide area network ("WAN") 202 is a system which supports distribution of application and data over the large whole geographical field. A transport network connects the IDNA node 204 based on an optical synchronous transfer network (Synchronous Optical Network) ("SONET"), and the application in those nodes enables it to communicate mutually.

0033] Each IDNA node 204 contains the intelligent call processor ("ICP") 172 and the resource complex 180 (drawing 1 ). Drawing 3 shows the IDNA node 204 which has the resource complex A("RCA") 206 and the resource complex B("RCB") 208. Those functions may be absorbed by the function offered by the network management system ("NMS") 212 although ICP can be connected with the auxiliary processor 210 which offers the existing support functions, such as facility offer, accounting, and restoration. However, in a desirable operation form, those support functions can be offered by the concentrated type service management (Service Administration) ("SA") system 500 which has the data control (Data Management) ("DM") component 400 so that this specification may explain in relation to drawing 4 (a). As further shown in the detail, ICP172 can be connected with other ICPs172, other networks (not shown), or other equipments (not shown) by the direct link 214 which has signaling 216 and the transfer link 214 at drawing 3 . A direct link abolishes the latency time between the connected equipment, and enables equipment to communicate in a self language. ICPs172 are the "brains" of the IDNA node 204, and it is desirable that it is the general purpose computer which can be crossed to a large-scale-computer network from one processor equipped with the storage by the requirements for processing of the IDNA node 204. A general purpose computer has redundant processing and memory storage and connection.

0034] A general purpose computer points out the thing of a commercial ready-made component contrastive with the equipment of the exclusive use which was specially constituted in switching of a telephone and was designed, or the computer which can be assembled by that cause, when used on these specifications. Integration of the general purpose computer in a call network offers many advantages.

0035] By using a general purpose computer, in order that ICP172 may fill a high processing demand, the function extended by additional hardware is given. Such an addition product includes the function which raises a throughput, a



ata storage, and communication-band width of face. Such an addition product does not need the software and/or the hardware change of each switch peculiar to a maker in a call network. Therefore, new service and a new protocol can be mounted and installed on a general scale, without correcting each equipment switching within the net. The change to the intelligent call processor 172 from the integral switch 20 ( drawing 1 ) provides this invention with the above advantage and a high function.

0036] In the case of the use which needs a higher throughput, the price / performance ratio of call processing can be optimized by multiprocessing using a cheaper processor. For other uses, it is advantageous to use more highly efficient equipments, such as a minicomputer which has higher processing speed, it is required, or cost efficiency is sometimes high.

0037] As stated above, ICP172 contains the cluster of a general purpose computer which operates on UNIX (registered trademark) or a Windows NT (registered trademark) operating system. For example, ICP172 consists of 16 32-bit processors which operate by 333MHz in a symmetrical multiprocessor cluster in the big application which supports a maximum of 100,000 ports with one resource complex. For example, a processor can be divided into four separate servers equipped with four processors, respectively. Each processor is a SA network (System Area Network) "SAN").

It connects with the clustering technology of \*\*\*\*\*. A processor cluster can share access to Redundant Array of Independent Disks ("RAID") modular data storage. The storage shared can add a modular disk unit, or \*\*\*\*\* adjustment can be carried out removing. As for the server in a cluster, it is desirable to share the redundant link to RC180 ( drawing 1 ).

0038] It is shown in drawing and ICP software architectures are (1) management software, (2) ICP application, (3) calculation hardware and software, (4) resource complex component, and a open-shop-operation model that makes possible (5) service architecture and compatibility of processing at a row like "plug-and-play" function of a personal computer. Such general-purpose architecture cuts down a maintenance cost with standardization, and offers the profits obtained by volume efficiency.

0039] Therefore, this invention enables division of development, and use of a modular tool, and, thereby, development and mounting of service accelerate it. Furthermore, the side relevant to use of service management is within the limits of the control of a network operator based on required criteria in contrast with the hardware supplied by the fixed messaging protocol or the given maker and the restrictions received with a specific combination of software.

0040] this invention can make the whole network distribute service and a function by use of a managed object flexibly in place to desire", and dynamically ("under execution") again based on the factor of arbitrary numbers, such as capacity and the amount used. In order to operate on the plat form where the service processing 22 ( drawing 1 ), the call processing 24 ( drawing 1 ), and the facility processing 26 ( drawing 1 ) are of the same kind, a performance improves. Furthermore, this invention enables the surveillance and processing of a call sub element which have not been used before. this invention supervises a function and service, and when they do not become outdated or stop using it, it can delete them again.

0041] The resource complex (System Area Network) ("RC") 180 ( drawing 1 ) is a set of the physical unit or resource which offers transmission, signaling, and a connection service. RC180 which can contain the intelligent peripheral device 88 replaces the switch mechanisms 28 and 158 ( drawing 1 ) of IN, AIN, or SSI/SF architecture. Unlike IN or AIN architecture, control of resource complex, such as RCA206, is a low. Furthermore, RCA206 can contain two or more switch mechanisms 158. A switch mechanism 158 and other customer interfaces (not shown) are connected to two or more subscribers and switching networks by standard telephone connection. Such a customer system can contain the ISDN terminal 52, facsimile equipment 220, a telephone 54, and the PBX system 222. ICP172 communicates with RC180 ( drawing 1 ), RCA206, and RCB208 through the high-speed-data-transmission pipe (at least 100mb [/second ] Ethernet (registered trademark) connection) 224. RC 180, 206, and 208 can be compared with a printer, ICP172 can be compared to a personal computer and a personal computer controls a printer here using a driver. The "driver" in the IDNA node 204 is a resource complex proxy (Resource Complex Proxy) ("RCP") (not shown) explained with reference to drawing 5 later. Thereby, a maker can offer an IDNA conformity node using this interface, without rewriting all software, in order to incorporate an IDNA model.

0042] Furthermore, level lower than what is generally offered by AIN or IN architecture has the resource complex 180 ( drawing 1 ) and control of RCA206 and RCB208. Consequently, a resource complex maker does not need to provide network owner with a specific call and processing of service that what is necessary is just to offer one interface, in order to support a facility and network administration processing. The interface of a low is extracted by more nearly individual operation. Since the number of interfaces is one, a network owner can choose from the resource complex of the large range based on the determination about a price and a performance. An intelligence is added to ICP172 instead of RC180, RC180 is separated from change, and the complexity decreases. Since the role of RC180 is simplified, a

change is made more easily and, thereby, switching of alternatives, such as an Asynchronous Transfer Mode (ATM), and the shift to transmission technology become easy.

0043] The intelligent peripheral device (Intelligent Peripheral) ("IP") 88 is equipped with the function to process the information included in an actual call transmission path, and to operate based on it. Generally, IP88 is in individual resource complex, such as RCB208, and is controlled by ICP172 like RCA206. IP can offer the function to process the data within a call transmission path on real time using digital-signal-processing (Digital Signal Processing) ("DSP") technology.

0044] A network management system (Network Management System) ("NMD") 212 is used in order to supervise and control the hardware in the IDNA network 200, and service. The embodiment of NMS212 proposed is good at the framework of the telecommunications management network (Telecommunications Management Network) ("TMN") conformity which manages the component in the IDNA network 200. More specifically, a network management system 212 controls expansion of service, maintains the health of those services, offers the information about those services, and offers the network level function manager of the IDNA network 200. NMS212 accesses service and hardware by the agent function in the IDNA node 204, and controls it. The ICP-NMS agent in the IDNA node 204 (not shown) performs the command or demand advanced from NMS212. NMS212 supervises RCA206 and RCB208, directly, and controls them by the standard operation link 226.

0045] As furthermore shown in drawing 3, the managed object environment (Managed Object Creation Environment) ("MOCE") 228 contains the subcomponent which creates the service performed in the IDNA network 200. The composition unit and API expression independent of the service used in order that a service designer may create new service are embedded at the graphic user interface ("GUI") which is the main subcomponent of MOCE. MOCE228 is the meeting with which the tool which works as a host of one user environment called service creation (Service Creation) ("SC") environment or a platform was unified. This is the meeting of the processing needed for the whole process of service creation, such as a definition of service documentation included in a managed object and a service examination, and a managed object, a definition of an interface, a definition of a protocol, and a definition of a data input. Since a managed object is applicable to all the nodes on a network, a network owner should just develop one service at a time using MOCE228. This is contrastive with the network owner who is not, if it is the hachures which various switch makers are alike, respectively, are made to develop service of the maker's version, and develop service repeatedly by that cause.

0046] MOCE228 and NMS212 are connected through a repository 230. A repository 230 contains the managed object which is distributed by NMS212 and used within the IDNA/NGIN node 204. Although a repository 230 offers the buffer between MOCE228 and NMS212 again, in order to perform the "student" network examination shown by the dotted line 232, the direct file of MOCE228 may be carried out to NMS212.

0047] According to the desirable operation form of this invention as shown in drawing 4 (a), an IDNA/NGIN system contains the concentrated service management (Service Administration) ("SA") component 500 which offers both storage (repository) 230 function and general-purpose network administration (NMS) 212 function of the IDNA system 170 with the added function. Generally, the SA component 500 as shown in drawing 4 (a) supports all services of an IDNA/NGIN system, the off-line storage of data, naming, distribution, activation, and removal, and offers further the data control ("DM") function which enables execution-time storage of the data used by the service object in an IDNA/NGIN service node, duplicate, synchronization, and use.

0048] It is the component which performs all functions more required for a detail since the service management component 500 constitutes both the hardware component which managed, memorized and distributed all the services and service data that are used by the IDNA service processing node, and was mounted in system IDNA/NGIN, and a software component as notionally shown in drawing 4 (b). As drawing 4 (b) was shown, generally the role of the SA component 500 Receive data from MOCE (service creation)228, receive the customer order data 502 from the legacy system 229 of an order received or others, and an IDNA/NGIN system is supplied for a customer's use. For example, into a service creation process, it responds at the demand for example, by the MOCE/SC E user. Data, the composition unit independent of service (Service Independent Building Block) "SIBB", A service logic program (ServiceLogic program) ("SLP") and other service logic components 503 For example, a service package of finishing [ develop to MOCE200, complete from MOCE228, and / an examination ], So that service logic or the data component 506, which are SIBB, SLP, and others, may be received, a name peculiar to each service component may be offered and it may explain to a detail more in this specification later It is distributing data and each service component 509 to the data control functional component 600. Furthermore, as shown in drawing 4 (a), the service management 300 maintains the repository 230 containing the global record database (Database of Record) ("DBOR") with which the data control component 600 contains all IDNA(s) services to which it receives and data for the data.

0049] The role of others of service management carries out activation of data and the service component 512. So that



ata, SIBB and a managed object, or the service logic program SLP may guarantee a bird clapper to a node possible use ] and may explain to it in detail through the data control component 600 later Data and the name of SLP and .IBB515 are registered by sending the logical name to the Network Operating System ("NOS") component 700 for registration. Data and the service component 518 are stopped and it includes removing data and service 521 from an IDNA/NGIN system through the data control component 600. Service management performs a configuration function manager further by maintaining the states (finishing [ a pre-examination ] finishing [ a post-examination ] finishing expansion ] etc.) of each SIBB and service other than revision by the naming process. Thereby, it is guaranteed that service is not developed until it examines all the components of the service with the sufficient result and can constitute item.

0050] As shown in drawing 4 (b) in more detail, the service management component 500 performs the function which constitutes and supplies the IDNA/NGIN service node 204 further according to the configuration information which SA receives. In detail, based on the received configuration information, the SA component 500 determines by which server that distributes the function of each component in each service node 204h, and which service and data to which node, or is in a service node to give one's service, and the cache of which data is carried out to the local-memory resident relevant to the IDNA/NGIN node server. In detail, SA develops the configuration rule included in the service profile (configuration) file 580 in the local (node) resource management ("LRM") component 575 of the NOS system 600, in order to memorize into the local LEN cache arranged at each service node. Such a configuration file 580 determines the service performed by the IDNA node. LEN reads first this service profile file 580 memorized by the local cache of the node, determines, specific service layer execution environment (Service Layer Execution Environment) ("SLEE"), for example, virtual machine, which performs service according to the rule in a service profile file, and determines the service in which performs actively (as a permanent object) in SLEE, or only a demand carries out instance generation.

0051] If drawing 4 (a) is referred to again, the NGIN data control component 600 will function in both the life cycle of service, and the use functional force of service. a service management component -- GRU -- in a \*\*\*\*\* case, the data control component 600 provides each IDNA/NGIN service node with a local data storage and a data control function or a BAL record data base (repository) This contains the data of all types including a service program and SIBB, the data for service (a customer profile, telephone number, etc.), multimedia files (voice file for interactive-mode audio responses (Interactive Voice Response) ("IVR") etc.), etc. Specifically, the data control component 600 of a service node receives the extract of SA global DBOR containing all data required for the service performed by local NGIN service which was specified by service management. This mechanism is explained more to a detail in relation to drawing 4 (c) later.

0052] In a desirable operation form, the data control component 600 of SA component provides each IDNA/NGIN service node with a local data storage and a function manager. In detail, data control memorizes the data received from service management to one or more data bases, by carrying out the cache of the required data on the database server arranged in the memory resident or the same place in a service control computer, can make service/data usable easily at service control environment, and, thereby, can provide service control service with service/data by the minimum latency time. More generally regardless of whether the data control component 600 was received from service management, or it received as a result of service processing, real-time storage, duplicate, synchronization, and use of data are performed. Such a DM function can be further classified as 1 data repository function, 2 data manipulation functions, 3 data use function, and a 4 billing-record generation function.

0053] Next, with reference to drawing 5 , the logical-function view of the communication system which uses the intelligent distributed-network architecture 200 by this invention is explained. ICP172 is shown that the ICP-NMS agent 240 and SLEE242 which process as a host various managed objects 246, 248, 250, and 252 obtained from the managed object base class 244 are included.

0054] Generally, a managed object is a method which mounts a software function, and each managed object offers both a functional interface and a management interface, in order to realize the function of a managed object. A management interface controls access to the people and the object which can use a managed object function. In this invention, all the application software for a telephone except the infrastructure software performed by the IDNA/NGIN node 204 is developed as a managed object and a support library. This offers the interface and mounting into which it was unified for controlling and managing IDNA node software.

0055] A set of the network element which connects, routes and carries out termination of the transmission traffic processed by the node is collectively called resource complex ("RC"). The resource proxy (Resource Proxy) ("RCP") 244 is used for the service processing application performed on SLEE as a control interface of RC180. RCP244 is connected with a device driver and can apply the command for which it does not depend on equipment from the object in SLEE to the command peculiar to a facility executed by RC180. RCP224 can be explained as an interface which

realizes a basic command common between the venders of the property in RCP244. RCP244 is realizable as one or more managed objects which operate on the IDNA node 204, as illustrated. As an alternative, this function can be offered as a part of RC180. NMS212, a repository 230, and MOCE228 are in agreement with explanation of those elements by consideration of drawing 3 - drawing 5 (a).

[0056] Note that the operation link 226 carries out the direct file of NMS212 to RC180. This corresponds to the more additional role of the network management system at the time of supervising the situation of network hardware of operation. This can be performed independently of IDNA architecture (using the well-known TMN technique). Furthermore, RC180 is connectable with other resource complex 254. Moreover, the direct signaling link 214 which goes into ICP172 so that the signals 216, such as SS7, can go into call-processing environment directly is shown. By interrupting a signal in the circumference of a network, SS7 message can progress to ICE172 directly, without passing along RC180. Thereby, a signal path is short, by the bird clapper, the latency time decreases and robustness improves. The transmission line 218 accompanying this connects with RC180.

[0057] Drawing 6 shows hierarchization of the functional interface in ICP172. MOCE228 is a system which generates managed object software and its follower. NMS212 controls execution of ICP172 by connecting with the agent function offered in ICP172 called ICP-NMS agent 240. NMS212 controls operation of the local operating system 260 in ICE172 (Local Operating System) ("LOS"). NMS212 controls operation of ICE172 including an inquiry of the start of a process, the contents of a halt and a process table, and the situation of a process, the composition of an operating system parameter, and the surveillance of the performance of the general purpose computer system which works as a host of ICP172.

[0058] NMS212 controls operation of the wide area network operating system (Wide Area Network Operating System) ("WANOS") 262 again. NMS212 is NMS to the control row of LOS260 about the composition of initial setting of a WANOS support process, operation, and a WANOS library. It controls through other interfaces offered by SLEE242. NMS212 controls one or more instance generation and operation of SLEE242 which operate on ICE172.

LOS260 is the ready-made operating system of marketing for operating a general purpose computer. WANOS262 is the ready-made middleware software package (for example, object requester broker) of marketing which makes seamless communication between calculation nodes easy. SLEE242 works as a host of execution of the managed object 244 which is the software instance which realizes service processing architecture. SLEE242 realizes a means to control execution of the managed object 244 by the ICP-NMS agent 240. Therefore, SLEE242 instance is a software process which enables a dialog with the NMS-ICP agent 240 at the time of realizing management of expansion of managed object software, the instantiation of deletion and the instance for management, deletion and the dialog of a managed object, support of cooperation, and access to the peculiar library (Native Library) 264, and required control.

[0059] The peculiar library 264 is a library coded so that it might be dependent only on LOS260 or WANOS262, and peculiar general purpose computer execution (for example, C library [ finishing / compile ]). These are used in order to mainly compensate the peculiar function offered by SLEE242.

[0060] The SLEE library 266 is a library coded in order to perform in SLEE242. They can access the function offered by SLEE242 and the peculiar library 264. A managed object 244 is software loaded and performed by SLEE242. They can use the function offered by SLEE242 and the SLEE library 266 (and ordinary library 264 probably).

[0061] The ICP-NMS agent 240 provides NMS212 with the capacity which controls operation of ICE172. The ICP-NMS agent 240 realizes capacity which controls operation of LOS260, composition and operation of WANOS262, composition, and the instantiation of SLEE242 and operation. The proposed service processing architecture operates in high abstraction layer. However, the perspective diagram of SLEE242 shows that there are only two layers, the managed object layer 244 which is an object layer (software instance) which has a dialog under control of NMS212, and the library layers 264 or 266 which are software layers (peculiar to one of SLEE242 and LOS260) which provide operation of a managed object 242 or SLEE242 the very thing with a supplementary function. However, it is expected at a certain time that NMS212 may abandon control of the exact place of a managed object instance. For example, a managed object instance is movable to another node from a certain node based on one or more algorithms or events which accept a demand.

[0062] Collectively, the function of LOS and WANOS should understand what can be expressed as the Network Operating System which carries out the work which offers the connect function which is not dependent on a platform between IDNA/NGIN system components, and is not dependent on a place, or "NOS", as shown in drawing 6. That is, LOS includes the service covering the process interface between other IDNA/NGIN functional components and a subcomponent, and 1 set of whole networks which offer communication. Between the services offered by NOS, object connectability, logical name translation, InterProcess communication, and a row have local and system wide resource management ("RM"). For example, as shown in drawing 4 (a), the NOS component 700 offers a local (NODE RM) and system wide resource management (SYS EM) function. In detail, an NOS component encapsulates the place of service

from the process which needs service and data, therefore a process should just call one logical name. Next, an NOS component determines which instance of service is used, and provides the instance with connectability. NOS700 makes possible partially both the property which IDNA/NGIN is large and was distributed, and the platform independence of IDNA/NGIN. For example, the above-mentioned logic program can call other logic programs using the NOS component 700, and can call and start other logic programs which operate on SLEE from which the same service node or a remote service node differs as a result. In detail, a service node can be specified that it performs only a certain service by SA500. For example, when a call reaches the switch which has the related service node 204 which cannot perform required service like connection of a conference bridge, IDNA may have to send a call to another node constituted so that such service might be offered. As for IDNA, it is desirable to provide with a service response the switch which calls the required service in another remote service node through the NOS component 700, performs call processing, and is in the original node.

0063] Next, with reference to drawing 7, the class hierarchy (hierarchy) of the object managed by this invention is explained. The abstract base class management object 244 includes the virtual function, in order to guarantee that common functionality and all the derived classes may be appropriately supported as an object in SLEE242. Especially, four different subclasses 252, i.e., a service control class, the call control class 250, the base material control class 248, and the object resource proxy class 246 are shown.

0064] The service control class 252 is a base class for all service function objects. The session manager class 280 is encapsulating session related information and activity. A session can do a bird clapper from the call or other appeals beyond 1 or it of a network function. The session manager class 280 has prepared the only identifier to each session. If all processing is performed by the intersection method, bill creation (billing) information must be collated in that case. The only identifier for [ each ] a call makes collating easy instead of needing collating processing high in cost. In service processing, the protocol is wrapped by the continuous abstract concept layer. After all, a protocol is fully abstracted so that assignment/instantiation of a session manager may be justified (for example, in SS7, reception of an AM message has and justifies session management).

0065] The base material capacity class 282 changes the quality of service on a base material. or the service control class 252 enables change of the quality ("QoS") of service of a call -- or -- the same -- movement in higher proportion from 56 Kbit/s -- and subsequently the base material capacity like descent can be changed Connection manager class 202 \*\*\*\*\* management of the QoS is carried out. For example, the half rate subclass 284 replaces QoS of a call with the usual sample rate of 8Khz(es), and reduces it to the sample rate of 4Khz. The stereo subclass 286 may permit formation of two connection of a call of a user, in order to support a left channel and the method channel of the right.

0066] The service Arbitration class 288 compiles allelism of service, and mediation of the interaction of service into one book. Since the service control class 252 may be opposed to each other with a start and end service, especially this is needed. It is not desirable to encode recognition of how to solve allelism with other molds of the service control class 252 in each service control class 252 for many practical reasons. Instead, when allelism is discriminated, the service Arbitration class 288 lets the reference to opposing services and those undecided demands pass. The service Arbitration class 288 can determine a suitable action course in consideration of the next question which probably receives a local background, configuration data, and opposing service in that case. Having the service Arbitration class 288 permits clear documentation which counters hard coding or an absolute mechanism, and coding of the algorithm of allelism solution. or [ that service is moreover updated ] -- or when added, existing service does not need to be updated so that change of all allelism that may need change of much relations within single service may be clarified

0067] The feature class 290 performs the standard set (for example, the 3 direction call, call waiting) of the capacity coordinated with communication by the telephone. Such one capacity can make generating of connection release of the call which is existing in order to reach the planned addressee the override 292 made possible. Other common capacity can include the call block 294, and, thereby, a start application may be refused based on 1 set of standards about a start.

0068] The service discernment class 296 causes service of the period throughout call processing others alternatively, and is classified into a low rank as service itself. The service discernment class 296 removes the need of having the mode fixed in each service for determining when it has flexible background sensing service activities and service is made working. The activity sequence is isolated from service itself. For example, Subscriber A and Subscriber B can access the feature of the same group. Subscriber A chooses so that more than 1 or it of service of him may be alternatively called using the signal of a specific group. As for Subscriber B, it is desirable to use the signal of a group which is different in order to make his service work. The only difference between subscribers is the way they make their service work. Then, it is desirable to divide the selection method from service itself. There are two solutions which can be used. In order that Subscribers' A and B service selection method may be encoded in the separate service discernment class 296 or the service discernment class 296 may direct suitable information, the profile per subscriber

can be used. This may be drawn so that it may be applied to many users rather than the service set is disassembled. Furthermore, use of the service discernment class 296 makes it possible to change mapping of access to service based on the given background of a call or progress. Probably execution of this class may make much call persons concerned use various services using various activity inputs. In the conventional technology, the inflexible service selection scheme from which all the switch sellers prevented this capacity was supplied.

0069] the storage and transmission 300 to which the media independent class of service 298 is applied to the mold of the various media containing voice, Fuchs, an e-mail, and others', and it takes in advance (pulley en PUSHON) and is in the form of the service control class 252 like QoS and \*\* person connection [ broadcast and ] If it is developed so that the service control class 252 may be applied to each media form, the service control class 252 may be pushed by the reusable service control class 252 in that case. If the service control class 252 is pushed by a media dependence function and the media independent function (namely, the media independent function SC perform service and 1per set media dependence wrapper SC-media form-), storage and transmission 300 are equipped with the comprehensive capacity that some the messages or data flow of media form are memorized, and capacity which subsequently supplies it to media form based on some events so that it may be derived from the media independent class 298. Directional change is equipped with capacity which moves connection to other logical addresses from the one logical address based on the specified conditions. This concept is the foundation, such as call transmission (all form), automatic call distribution/UCD, WATS (one to 800 service), find me / follow me, and a car telephone. It negotiates, or by other methods, call waiting, priority prefetch, etc. solve prefetch and it includes service. QoS modulation connection performs service of further others over voice/Fuchs, the video that is flowing, and the packet network like a file transfer. \*\*\*\* connection includes three directions, the direction video meeting of N, etc. Although a user controls and an input is mainly performed using the key of a telephone, it is expected that speech recognition is used for user control and an input in the future.

0070] The connection manager class 302 is [ of coordinating and arbitrating connection of the various base material control 248 followed on a call ] responsible. Therefore, it dissociates from other the services of all, and the complexity which manages much connectability between the participants under call is removed. Joint release of service and the call processing is carried out from connection. This destroys the paradigm of the mapping call to connection like connection of 1 connection pair large number. Now, much mapping of a call pair call is opposite a large number.

0071] The connection manager class 302 in architecture is designed so that it may operate independently (stand-alone), or so that it may cooperate as an equal person. Setting to an operation, the service control class 252 shows the connection manager class 302 by the demand which adds, changes and removes a call segment. The duty of the connection manager class 302 is making and accomplishing these change. Since, as for minding, connection may be considered as an attribute of a resource as a resource of these very thing, the connection manager class 302 may be performed as a resource function manager's basic proxy or basic mode.

0072] It specifies how the call control class 250 performs indispensable call processing like the fundamental finite state machine used in common with communication by the telephone, and call processing is performed. Two classes may be drawn along with the functional partition of start (a call is arranged) 304 and end (a call is received) 306.

0073] the base material control class 248 -- the resource complex 180 -- and it is turned to fitting the specific signal and specific event of that to the common signal and common event which may be understood by the call control object 250 via the resource proxy 246 A subscriber line number, the class of service, the form of access, etc. dispel one role of the object drawn from this class expected, and it is to collect the information about the start edge of a call. A subclass can specialize on the basis of the number of the circuit coordinated with signal transmission, or channels. These can contain the channel [ which is shown by SS7 signal transmission 318 completely separated from the channel related class 308 which is applied to a single signal transmitting channel per 23 base material channels under ISDN primary interface 310, the channel single class 312 which is type-ized by the analog telephone 314 used for calling to controlling a single circuit, and the base material channel ] common class 316.

0074] It is turned to the resource proxy class 246 interfacing the operation environment over the real world switch and other elements in a base material network. The example of the internal state which is performed on this level and inherited by all inherited classes is [ in-service pair AUTOOBU service and free opposite ] under use. The meant inherited class is the modem connection 326 (standard proxy to a digital modem) corresponding to the resource of the special mold in a telephone 320 (standard proxy to standard 2500 sets), the audio response unit ("VRUs") 322 (standard proxy to an audio response unit), the IMT trunk connection 324 (standard proxy to a digital trunk (T1/E1) circuit), and the resource complex 180. The suitable method which may be useful to the service request at which a service control component arrives is explained further with reference to drawing 10 below. Drawing 10 shows specially other operation forms of the service control environment 430 where it has SLEE use 450,450' currently carried out within the service control server 435, for example, the operation system of a general purpose computer 440.

0075] As shown in drawing 8 , SLEE450 is the Java (trademark) "virtual machine designed so that the logic program object) of at least five molds performed although call processing service and other support services are offered might be carried out." Namely, the 1 feature discernment logic program 510 "FD." This is the functional subcomponent of the service control class / service discernment class 296 ( drawing 7 ) which calls other suitable service logic programs for receiving a service request from a switch platform first, and determining the service which should call based on the standard in which some use is possible, and subsequently processing a call. 2) Service logic program ("SLF") object 20. This is the functional subcomponent of the service control class 252 ( drawing 7 ) which performs received service processing for a service request or events. 3) Line logic program 530 "LLP." This is the functional subcomponent of the call control class 250 ( drawing 7 ) which maintains the present state of a network access line. 4) Event logic program ("ELP") object 540. This is the functional subcomponent of the service control / session manager MAKURASU 260 ( drawing 7 ) by which all other logic programs write an event in it. 5) Call logic program ("CLP") object 545. This is functional subcomponent \*\*\*\*\* of the service control / connection manager class 302 ( drawing 7 ) which maintains the state of the whole call by preparing the node for all the logic programs of the others followed in processing of a call. or [ that each of these logic programs is temporarily illustrated so that it may be explained ] -- or it is preferably realized as continuable software "an object" written to Java (trademark) programming language. These objects are written in MOCE/S C E only at once, and IDNA/NGIN service control architecture is designed so that it may be developed by SLEE anywhere in a network about all types of computer, and all types of starting device.

0076] FD510 is a static subcomponent which determines that the information which receives the service request from the resource complex switched when discriminating first that a switch is processed according to the bigger feature and service is processed by IDNA/NGIN, and is related with; 2 service request can be analyzed, and; and 3SLP can process service request. Preferably, although FD includes the called number, the called number, the start switch ID, the started trunk group, the started line information, and the network call ID, it may be the system task or the illustrated object for receiving the data supplied from the resource complex which is not limited to them. By NOS, FD510 begins instantiation of suitable SLP and LLP CLP(ed) and started, in order to process a call. Preferably, FD510 is an object which is not combined with a specific call or an event and to continue, and is operated actively [ always ] in service control SLEE450. In order to divide a load and to guarantee real-time efficiency depending on the amount of the demand to the complexity of the analysis carried out, and FD, the stage beyond 1 or it of FD currently actively operated in service control SLEE450 may exist. For example, while it can be used for one FD analyzing SS7 received message data, it may be used for FD of another side analyzing ATM message data.

0077] the state of 1 network access point, connection, or a line present in the line logic program (LLP) 530 -- maintaining --; 2 -- it is the functional subcomponent which questions data management about the feature coordinated with the physical point, connection, or a line, and applies those features like overflow routing as; 3 call interception and call waiting, call transmission, and a call position demand There is "LLPT" by LLP coordinated with the line which starts a call, LLP coordinated with the line which "LLPO" and point connection, or a call ends below, and the following. Once a line logic programme step (instant) is illustrated, itself will be recorded on switch structure. The line logic program 530 is transmitted to the ELP subcomponent of the same stage of service processing of all event data so that it may be explained.

0078] Dynamic subcomponents are those components destroyed when it is dynamically constituted according to the various stages of service processing and the stage of service processing is completed, and include event logic program (ELP); call logic program (CLP); and the service logic program (SLP).

0079] The event logic program (ELP) 540 is a functional subcomponent used for holding the real-time event data which record all the event data that are generated throughout service processing and generated throughout implementation of service. Preferably, an event logic program is illustrated by call control processing with a switch, when an event is received first. When a switch transmits a service request to NGIN, the demand passes in accordance with the address of ELP so that it may be transmitted to this logic program with which event data were combined with the call. The event logic program is accessible to all the subcomponents in the same stage of service processing, i.e., LLP relevant to a call, LLPs, and SLP. Since the call is processed in execution of each service processing component for service, according to the rule established beforehand, event data are written in ELP by NOS. If a call is completed, the event data in ELP are written in data storage or a log, and from this data storage or a log, subsequently to billing record event data will be compiled, and they will be sent to the system of the lower stream of a river for a billing, the amount of telephone calls / use report, and other back office functions. Especially ELP carries out the function to carry out formatting, to carry out; 3 check of the; 2 event, and to permit [ collect the network events generated by the call of 1 specification, ] it to suitable call history record ("CDRs"), for example, call detailed record, billing data logging "BDRs"), switch event record, etc., and to memorize information to data management for future transmission to a



low-stream system, for example, customer bill creation. The rule which determines that an event is written in ELP what should be understood is established in service generating. Event data are additionally accessible by flow former crew MENTO and the network system.

0080] The call logic program (CLP) 545 is a functional subcomponent which maintains the state of each SLP followed on service processing, and forms the process interface between all services (LP's). In 1 operation form, CLP may be illustrated with the call control component which was illustrated by FD or was put on the switch, when an event service demand is first received about a call. In alternative, CLP545 can be illustrated by SLP510 in the point of period some throughout service processing according to the trigger point programmed by SLP, and instantiation of CLP may be peculiar to service in; this gentleman method. A call logic program receives the address of all subcomponents, i.e., SLPa, LLPs, and ELP within the same stage of service processing in the time of instantiation. Subsequently, CLP associates SLP, LLPO, LLPT, and ELP (s) about the call, and is accessible in all these subcomponents within the same stage of service processing. If a call is completed, CLP will be notified to all the subcomponents within the same stage of service processing of the completion of a call which starts the decomposition process of a logic program.

0081] The service logic program (SLP) 520 is a dynamic subcomponent which supplies the logic needed for giving one's service. It is rather combined with service from a call, and SLP performs the feature about a call included in service and it. The feature by which SLP may be applied to service includes for example, a call routine algorithm and VR service. About the service which can make it the object which the service used frequently continues, or is required by FD and used frequently, for example, SLP may be illustrated, when stopped at the time of the completion of a call. The configuration file 580 in which fixed SLP is generated by service management about the service as it indicated to be to drawing 11 whether only a certain time or a demand is activity always is determined. Preferably, SLP accesses CLP and ELP within the same stage of service processing.

0082] It can use for the task which no SLP is related with specific call service, and some SLP is needed by other SLP, and is called by it. In order to follow, for example, for SLP for 800 service to complete the task for call routine relay, it may be necessary to pull out SLP for a line information database question. Moreover, SLP can let control of the call processing for a call to other SLP pass. Preferably, one control SLP should be carrying out at the time of the single stage of service processing. All the event data generated as a portion of the service task performed by SLP are sent to the ELP component 540 within the same stage of \*\*\*\*\* processing.

0083] Since all the information on the operation system which this should carry out is not included, SLP cannot be directly carried out in an operation system. It is prepared, in order that the NOS middleware between SLP and an operation system may maintain coincidence of SLP which crosses an operation system, if it is necessary to carry out in various operation systems, without moreover SLP changing a format and the content.

0084] As further shown in drawing 8 , other processes carried out within SLEE450 about support and an operation function It de-energizes [ carry out the load of the service operated in SLEE, and it is activated, and ] and removes. And all services of the others operated within the SLEE are supervised further. And although it interfaces with the service manager ("SM") object 554; NOS service with the responsibility of reporting a state and use data to NOS The NOS class library used for being used and calling NOS service by all services operated within SLEE, that is The NOS client process 558 which is a gateway to NOS; in order to carry out simultaneously, without combining all SLEE resources Thread Manager ("TM") 557; and drawing 4 (c) equipped with the functionality needed for NGIN service are referred to. The data management ("DMAPI") API 410 used for interfacing with the local memory 415 of DM400 and memory manager component which are explained in this book is included.

0085] Other service stages include the Thread Manager stage 557 coordinated with the stage 559 used for service activities in a service node which is further explained in a detail in the service agent ("Sag") stage 559 and this book in the \*\*\*\*s which carry out a load to SLEE as shown in drawing 8 .

0086] Drawing 12 (a) shows the process step which prepares a mainstream ON point in a SLEE process (SLEE. Java). as shown in drawing 12 (a), Step 602 DM system can be used. In order to be used for interfacing with the NOS client process 558 and NOS service and to call NOS service It can use for receiving a name with a logical NOS site locator (positioning) system including the NOS master process of preparing the NOS class library used by all services currently operated within SLEE, and object reference registration. And the service control server operation system NT, UNIX, and PC, for example, the WIN douse etc., is considered starting a SLEE process or that it can do by recognizing the bootstrap call like main () or fork (). I hear that the NOS master component 560 ( drawing 8 ) interfaces with the operation system of a computer, the NOS client process 556, and other system components 571 directly, and what should be understood has it. desirable -- every -- the NOS master process 560 placed on the network containing all the NOS class libraries for interfacing with the NOS client object 558 on SLEE, and supplying NOS service or the local code exists Next, in Step 604, the hash table containing the key value pair as described a file which forms a shape

ject and shown at Step 606 can also be included by the service control configuration file. SLEE -- two parameters, e., a name, -- and configuration file \*\*\*\*\* is carried out A name parameter is a peculiar NGIN name string used by SLEE, in order to be used by the NOS locator service for discriminating this stage of SLEE, namely, to register itself by NGIN locator service (Step 612), and it is used by locator service in order that a configuration file may find the site locator. For example, it may be used for this table finding out a SLEE shape property. When NOS performs CORBA, ext, base CORBA functionality is initialized in Step 608. Next, in Step 610, as a SLEE class loader class is illustrated and NOS locator proxy service is shown by Step 612, it is illustrated within SLEE. Next, as Step 615 is shown, via a class loader class, the load of the service manager (SM) class is carried out, and it is illustrated, it is combined with part NOS, namely, SM object is registered by the local proxy NOS locator service object. I hear that what should be understood is spread to the service manager registration to other locator services in a NGIN field, and local locator service has it. /to SLEE after the service manager object was registered by locator service so that it might be explained with reference to drawing 12 (b) -- the service from it can be de-energized [ it can carry out a load, can be energized and ] and removed It is the SLEE thread which may make SLEE process an NOS event finally so that a process event loop may be carried out, and this loop may hold SLEE operation, and an NOS event may be explained more to a detail by this book and it may come through a service manager ("SM") and a service agent ("Sag") object, as Step 618 is shown.

[087] The process step carried out according to the service manager object stage 554 illustrated to have argued about drawing 12 (b) above with reference to drawing 12 (a) and Step 615 (service manager Impl. Java) is shown. Preferably, SM object performs the ORB interface for performing a service management operation for NOS. The process shows the step which is SM stage so that may carry out the load of (the service in SLEE, for example, a (load), and operation), and (start) (halt) the method, they may be energized and de-energized, and it may operate and it may end which is therefore taken. a \*\*\*\*\* [ that, as for the parameter which SM object stage let pass by NOS, the logic criteria and NOS of service for which it asks should register service by the NGIN local resource manager (LRM) site locator ] -- or the Boolean flag in which it is shown whether there is any responsibility which service registers in itself by NOS is included As shown in Step 620, the demand which carries out the load of the service is received first, and proxy naming service is processed at Step 622. Subsequently, judgment is made about whether the object which has materialized whether the load of the registered service, for example, one to 800 correction, ("18C") has already been carried out at Step 624 and the demanded service is illustrated. If the object about the demanded service is already illustrated, in order that NOS may arrange a physical object stage at Step 626 in that case, it will return to the object criteria of the service, and a process will return to Step 632. In order to carry out the load of the demanded service object for service, for example, all the classes for which the demanded service as which 18C is not already illustrated, and in which the class loader class contains other SLPs(es) and SIBBs(es) if it becomes depends in that case, it is illustrated in Step 625 which performs an inductive load. The inductive load is possible by mentioning a local configuration file from local memory. It lets the flag in which it is shown whether a class loader carries out a load to VM inductively especially in all these classes that depend pass. When carrying out the load of the class for service in the 1st phase, I hear that the load of it may be carried out if the load of it is not carried out yet, and being understood as a general service agent class. Subsequently, after carrying out the load of all the classes at Step 625, a Boolean register flag is checked at Step 625, in order to judge whether service must register by local NOS naming service (proxy) in itself. If a Boolean register flag is set as accuracy, service has the responsibility that NOS naming service is registered, in that case, as shown by Step 630. Otherwise, cooperation is made by continuing a process to Step 632 as which the SAg class is illustrated by letting it pass in a service agent stage in between the service agent object stage 58 (drawing 11) and specific services (i.e., an SLP object). Subsequently, in Step 635, it is caused, namely, a SLEE thread is coordinated with a service agent so that a new SLEE thread may be made in a method which is indicated and SLEE thread may operate a service agent. Finally, SM process is excited and a process returns to a SLEE. Java process. It has the responsibility of supervising all services of the others currently operated within the SLEE, and reporting a situation and use data to NOS, via the method formed in SM.

[088] Quotation of (SLEE class loader . Java) is explained more to a detail in consideration of drawing 12 (c) about SM process. Especially a SLEE class loader class is a class by which the class loader class of JVM was specialized, and is extending it. It is extending action of a system class loader by permitting a class by which a load is carried out over a network. Therefore, a class loader first checks the local memory coordinated with the stage of SLEE so that it might see, when the load of the class was already carried out and it was defined as the 1st step 686 of drawing 12 (c). If the load of the class is already carried out, a process will return in that case. In Step 688, in order that a class may check a local data storage (DM) if a message can use for that the load of the class is not carried out carrying out a load via NOS in that case if it becomes, it is sent. For example, although a SLEE class loader can search a class from the database of relation using JDBC database connectability, from the database of all the relation that supports JDBC API, I

ear that being understood can search a class and it has it. If a class of service is not found out by the local data storage, SLEE class loader will check a local file system in Step 689 in that case. If a class is found out by a data storage or the local file system, a class is fetched as Step 690 is shown. Subsequently, in Step 694, it is caused so that the defined class method may make the class which can be used for JVM operation environment. Especially a method (defined class) passes through each of the class specified in order to offer the service inductively, and changes a byte's array into the stage of Class Class. Subsequently this newly defined class may be made in Class Class using the new stage method. This functionality carried out the load of the SLEE, illustrated new service, and still remains generally. Preferably, as shown in Step 695, a method is searched for so that the POPYU rate of the local memory may be carried out, as the next time when the load of the class is carried out has a memory hit.

0089] a suitable operation form -- setting -- each of these illustrated objects -- general -- the following strings -- namely, -- ... site level, a SLEE number, and SLP name

the naming agreement which is alike and is therefore illustrated generally -- following -- the NOS locator service 577, e., LRM, -- them -- the very thing registration is carried out In this case, specific SLEE as which, as for the information; SLEE number concerning [ site level ] the NGIN service control server's 440 physical position, the object /as illustrated, for example, SLEE#1;, and an SLP name are the local names of service, feature discriminator #1.[ for example, ]. A string can contain a "version number" like a value. By a registration name being spread to other locator sites in a NGIN field, being able to; Come, and boiling it, and their being a registration process and NOS resource management functionality (as [ indicate ]), and knowing an NOS component by that cause more, when they are developed, as for those processes, service is easily developed, when it can use.

0090] The method and constructor of an object who were made with the class loader can refer to other classes. In order to determine the class referred to, a Java virtual machine calls the load class method of the class loader which made the class first. If required only in order to judge whether it exists in order that a Java virtual machine may reach whether a class exists and or not ] and it may know the superclass, ""determination (RIZORUBU)" flag will be set up imperfectly. However, if the stage of a class is made or either of the method is made, a class must be determined again. In this case, a determination flag is set up correctly and the determination class method is called. This functionality guarantees that the class / SIBBs / Java Beans referred to by service are determined with a SLEE class loader again.

0091] Drawing 12 (d) shows the service agent class process stream at the time of instantiation. As shown in Step 639, the 1st step includes instantiation of the Thread Manager ("TM") object which coordinates with a service agent and is drawn on drawing 11 as a TM object stage 557. As indicated, the Thread Manager object has put the foundation on the class which may be illustrated to act like the thread factory which is functioning as making a new SLEE thread for every service request (Thread Manager), or the thread wear house for which it asks when operating by the machine equipped with high thread generating potential capacity. Subsequently, now, it prepares for receiving the call event which SA coordinated with service goes into a process event loop via the (operation) class method, and is coordinated with service in Step 640.

0092] if drawing 12 (e) is referred to -- the (start) -- and (continuation) (end) the detail of the service agent who prepares a gateway in NGIN service via the class method is shown Each service in SLEE has the service agent object of the relation which has put the foundation on the class with the responsibility of managing a service stage (call stage) and sending an event by express to a service stage. As shown in drawing 12 (e), after illustrating a SAg object by the service manager (load) method and operating, the method (start) of SAg is caused for every new call which is requiring that service should be received. As especially shown in drawing 12 (e), in Step 641, the SAg start method first lets the flow of a message including the event information related with service processing for the call which is prepared by the initial address message ("IAM") from IDNA/NGIN mentioned in this book as for example, Ted (tid), an ORIDDO (orid) call identifier parameter, and a NeXT generation switch ("NGS") pass. Subsequently, in Step 643, the decode of the message flow is carried out by causing a method so that the important information related with the service stage may be extracted for example, (decoding). In addition, it is made so that the message information by which the call background object stage used for managing call background data was extracted may be received. As shown in Step 645, in the start method, a thread will be pulled out from the pool of a thread by being assigned to the call by causing now assigning a Thread Manager stage, if some of the threads for difference-\*\*\*\* are illustrated exceeding time, so that a new thread may be indicated by this book with reference to drawing 12 (g). If the SAg (continuation) method is caused by other methods, the object criteria corresponding to the thread assigned to the call will be returned.

0093] More nearly especially a Thread Manager object is desirable and the foundation is put on the Thread Manager which manages the thread which placed the foundation at Session ids. It is prepared, respectively, two methods and assignment) (release) in order to assign and release a thread. Both assignment and release expect the identifier peculiar as a key which may be used for thread discernment. It is used for the peculiar identifier including the object criteria ID ("Orid") which discriminate the processing ID ("Tid") set up by the NGS switch which received the call, and call cage



INETA, and discriminating a call stage. Drawing 12 (f) shows the detail of an operation of the method (assignment) of thread manager class. As shown in drawing 12 (f), in Step 660, a process lets Tid and the Orid identifier for discriminating call processing peculiar pass, and a peculiar key puts the foundation on an identifier and they are generated. Subsequently, in Step 662, when a question checks the hash table (medley table) of a key value [ for example, ] pair, a key is made about whether the thread which has already existed is discriminated. If a service thread means already having been assigned to the call and a key is accepted, after Thread Manager takes a hash table into consideration in Step 664 in that case, it will return to a SLEE thread stage (thread object). The increment of the counter which is in agreement with the number of the service threads illustrated in Step 663 by other methods is carried out, and determination is made about whether the maximum of the thread stage of the service exceeded at Step 665 in efforts which supervise a system load. If the maximum of the thread stage for the service is exceeded at the time of comparison with a counter value with the maximum service stage found out by for example, the service configuration file in this case, the step 667 -- setting -- a message -- for example, it is emitted to NOS so that other stages for service which can be used in other SLEE(s) which carried out to the same site or were illustrated in other service node positions can be searched for, and a process returns. As a SLEE thread instantiation process is indicated by this book with reference to drawing 12 (g), it is initialization of the priority event queue. If it becomes, in Step 668, judgment will be made in that case about whether the threshold of the thread stage for the service exceeded in the maximum of the thread stage for the service not being exceeded. If the threshold of the thread stage for the service is exceeded, warning will be emitted in that case by the NOS local resource management function in which the service threshold has attained. Finally in Step 670, the demanded new SLEE thread for service is assigned regardless of the output in step 668, a priority event queue is illustrated about the demanded service, and the control by which the thread is returned to the SAG stage for the service begins.

1094] After it returns to the service agent (start) method functionality as shown in drawing 12 (e) and Thread Manager assigns the thread for service stages, the object variable relevant to a thread is initialized at Step 646, and it is illustrated when the demanded new object stage of service causes a method (clone). Subsequently, in Step 648, the cloned new SLP stage is set as the newly assigned thread. Subsequently, in Step 650, judgment is made about whether the event information for which to coordinate with the call stage is needed, for example, all the IAM information extracted from the incoming message flow, exists. If there is event information coordinated with newly cloned SLP, as shown by Step 652 in that case, it will extrude on a thread. The thread newly assigned to the SLP at any rate is started, and whether event information which is extruded on a thread exists is waiting for asynchronous attainment of the event information on the service relation processed by the SA (continuation) method. As stated, the SLEE thread assigned to the call is maintaining the priority event queue, in order to hold the event information on all the service relation received throughout processing. All the events of relation have the priority of relation in service processing, and a thread manages processing of event information according to the position in the priority, i.e., the service event queue. Finally, in Step 654, a thread event loop is started about the call stage.

1095] I hear that the SA (continuation) method of what should be understood is the same as the method (start) shown in the real above figure 12 (e), and it is, and the difference is turned to carrying the event of real-time service relation with the service process thread already illustrated about the call stage as it argued above with reference to drawing 12 (e). Therefore, the service agent continuation method assigns again the service thread which received the event and discernment parameter of a call stage, and was coordinated with received Tid for events, and the Orid parameter, and extrudes an event to the event priority queue of a thread. I hear that both SAG and SM class consist of an IDL interface to NOS, and what should be understood has it. Although service (SLs) does not have such an interface, it can meet together and can open the SAG interface for free passage widely with a system. If the event which arrives at the stage where SLEE450 interpreted the following, i.e., directions throughout [ SLP and SIBB ] 1 service processing, and; 2 SLP is generated.; 4 SLP, SIBB, And the trace turned on on SLEE level It approves. and trace data It sends to the specified output.; 5 SLEE use data It generates. and operation-time use data It sends to the specified output, the exceptional data or; 6 telecommunication management network (TMN) interface are generated, and the performance data for a; 7 TMN interface are generated.; 8 SLP or the new stage of a practical use program It can perform supporting the same service in addition according to the multiplex service control stage for; and 9 load sharing, without receiving the message/demand for adding, and interrupting and deteriorating service processing in new SLP like a parenthesis, or a practical use program.

1096] When a service stage ends processing, it is open for free passage with the end of service, or service, and other processes are started. In the case of which, the SAG (end) method is called, and this functions as ending the thread stage coordinated with the call. This can be made and accomplished by causing the Thread Manager (release) method, and extruding through and an event on a thread event queue to Tid and the Orid identifier which discriminate a call

tage peculiar, releasing a call, ending a thread stage, and/or arranging a thread stage to a thread pool.

0097] Preferably, a SLEE thread class stage is equipped with the functionality needed for IDNA/NGIN which is carried out simultaneously, without combining all SLEE resources, and makes collaboration resource division easy. Mapping of 1 to 1 exists between a SLEE thread and a service stage by SLEE which coordinates one stage of service, and one stage of a SLEE thread especially, namely, there is a stage of the SLEE thread coordinated with a call about each call dealt with by service. A SLEE thread acts like the data warehouse for service by containing both Processing id (tid), the object criteria id (orid), object criteria, for example, an equal person, and an agent, and the priority event queue coordinated with SLP and SLP again. Especially a SLEE thread acts like the event channel between service (SLP) and a service agent by performing the pull supplier which enables push consumer [ for enabling two key interfaces, i.e., a service agent, to extrude an event on a SLEE thread ], and service to pull an event from the thread of those relation. Each SLEE thread has the stage of the priority event queue for putting a NGIN event into a queue in the indicated method so that it may be indicated.

0098] Preferably, a class (priority event queue) is a platform independent class which puts the event (inherited class of NGIN event) coordinated with service (SLP) into a queue. As shown with reference to Step 667,670 of drawing 12 (f), each SLEE thread object has illustrated the stage of a possible priority event queue for the bird clapper from the ash table of an event. In downward ranking, it can be put into an event by the queue, and event priority is defined in the NGIN event, and it is in the range of 10 to 1 anyway, and 10 is the highest priority. Therefore, each thread is made that use to processing is possible / pursuing the number of the events which cannot be used, or ], and enables perfect service processing coincidence in this way.

0099] Drawing 12 (g) shows how (post event) to encapsulate logic, in order to check supply of the event to the priority and the priority event queue of the event received by the thread, as shown in Step 675. As shown in drawing 12 (g), this compares the priority of the following queue with the priority of the extruded event on a priority queue which is processed at Step 678. It judges, when larger than the priority of the following event in a queue by which the extruded event will be processed at Step 680 (supposing it is). And [ whether the event extruded by the crowning of a queue so that it might be set up as a following event which should be processed as shown by step 682a is arranged, and ] Or as it is made a loop through a queue and shown by step 682b, by determining the position of the queue with which the event should be memorized according to the priority, it makes on substance and can accomplish. Subsequently, in Step 684, when the processing time is assigned from SLEE thread \*\* and a system, the next event of the highest priority is processed.

0100] More, a pull supplier interface is performed by the detail by the SLEE thread, in order to support the work for a customer who demands data from a supplier by causing the "pull" work which intercepts until event data can use or an exception occurs, and returns a customer event data, or the "try pull" work which is not intercepted. That is, if event data can be used, event data will be returned, and a husband event (hasEvent) parameter will be set up correctly, if; event data cannot be used, a husband event parameter will be set up imperfectly and a value equal to zero will be returned. Therefore, a SLEE thread acts as an event supplier, and service (SLP) takes over a customer's role. Since event data are fetched from a SLEE thread, service (SLP) uses a SLEE thread pull or a try pull. It is continuable without event data, if there is no service, it will use pull work, otherwise, it uses try pull work.

0101] A push consumer interface is a general push consumer interface which supports the work for suppliers for communicating event data to a customer by a SLEE thread's performing, causing the push work to a thread, and letting event data pass as a parameter at the thread priority event queue. Therefore, the SLEE thread acted as an event customer, and the service agent has taken over the role of a supplier. The service agent is using SLEE thread push work, in order to communicate event data to a SLEE thread. A bird clapper can do a "kill (kill)" service event from the highest priority. The priority of \*\*\*\*\* of an event is good, or when newly made IBENTOKURASU is designed, it may be established in service creation.

0102] They are all event \*\*\*\*\* to the service thread stage where the service agent stage for specific service was made for the call as indicated that /Swerved, and were received and generated in the process of service processing of a hell. For example, a bird clapper can do the first event generated by the switch in the node from a (service request event). This class the first service request to IDNA/NGIN service control to and profit The number of those to whom the port ID; call to which the switch ID; call by which the time; demand which a service request can begin is started from it is caused is caused and who are terminal facility ID; Calling; there is responsibility which the number of those who were called etc. solves and carries the initial call background information on relation. Time for connection to generate the subclass which is extending the NGIN event (connection event); the station number to which the number currently called is connected can be reported, and a \*\*\*\*\* virtual path ID and virtual path ID s to leave can be reported in the background of; and ATM-VNET service. The subclass which is extending the NGIN event (release event) can report a release event. For example, in the background of ATM-VNET service, release may be generated,

when those who were called or it was calling end a call, or when the user credit has run out. Time when, as for such a class, a release event is generated; SIBBS for determining the duration to the time when a release event is generated from the cause of generating a release event, and those who are calling, and who were reached and called can be performed. It may be used for the subclass which is extending the NGIN event (end event) sailing an end message from NGIN to NGS about this. A switch can start a decomposition connection process for this message at the time of reception. (Monitor release event) It is used for a subclass sending the message to NGS which has turned to NGS so that a NGIN event may be extended and release directions may be sent to NGIN at the time of reception of release directions. When NGS receives a monitor release message, a subclass (uni-no TIFAI event) sends the notice to cage INETA (call person), and may be caused. (Monitor connection event) A subclass extends a NGIN event and is a subclass used for sending a message to NGS from NGIN in order to send an event to NGIN, when a connection message is received.

0103] As indicated, in the context of real-time service processing, data management data retrieval and the updating function include capacity which accesses the data memorized by the period DM throughout service processing.

0104] In a suitable operation gestalt, DM receives the data demand from the managed [ execution ] object instance in SLEE through NOS during service processing by a certain specific service node. Data control (Data Management) is notified to especially a requiring agency (for example, managed object), when the data demand cannot be understood. When this data demand is a thing to reference of a data entity, data control is returned to demand-demanded data origin (minding NOS). In addition, ranging over two or more RIPOJITORI, data are supplied to operation and a support required in order to ask with DM among single RIPOJITORI. Furthermore, data control supports the collection or contrast of a result of the inquiry over two or more RIPOJITORI. When DM cannot pinpoint the place of the demanded name of an entity during a data retrieval demand, DM is notified to an NOS component. Moreover, an NOS component is notified when a database obstacle occurs during reference of a data entity. Furthermore, data control notifies the support which cannot search a specific data entity from an effective name to a requiring agency (a service control object is performed). When a data demand is a thing to renewal of a data entity, data control determines whether to update a data entity and a duplicate is required. DM is notified to a requiring agency, when the data entity which was data requiring and was specified cannot be updated, and further, when the place of the demanded name of an entity under renewal demand of data cannot be pinpointed, it is notified to NOS. DM is notified to NOS about the database obstacle at the time of renewal of a data entity during NGIN operation. As for DM, a data demand deletes the data item, to deletion of a data entity if, and it is determined whether initialization of a transaction is required on other RIPOJITORI.

0105] Drawing 4 (c) is illustrating the functional architecture of the data control component 400 as a whole. The data control component 400 is equipped with the database component 407 which stores and distributes the selected subset of the data which are embodied as the service control server component 405 with which call to real-time call processing, and it is made for service data to become usable by the service node, and a; individual database server, and are held by SA. Especially the service control server component 405 In order that it may be linked with the data control (DM) client 409 and; DM application which are actual data control application and DM application may gain data from SA. A DM which is the interface to be used API410 and; local cache strategy are followed. The local cache 415 which is a shared memory on the service control server used since all [ from usable DBOR extraction ] the all [ some or ] are stored to call processing. By implementing local cache strategy, the state of a local cache is maintained and it has the cache manager 420 who communicates with DM server and searches data from DBOR extraction. The database component 407 At the time of the service execution by the node concerned, by the managed object instance The data used One or more databases which it has The DBOR extraction 427 and; which are included The selected subset of the information which SA holds The DBOR extraction manager 426 who manages, and; The data received from service administration The data extraction from DDAPI424 and; DBOR extraction manager 426 which is a process interface between the SA client 422 and the; SA client 422 which are inputted into the DBOR extraction manager 426, and the data distributed processes of SA It has the data control server 425 dealt with generally.

0106] Here, data control operation is further explained to a detail about drawing 4 (c) and drawing 8 . Some kinds of functions may need managed objects (SIBB, SLP, etc.) and not only NOS but the data from data control 400 in SLEE. The each is illustrated to drawing 4 (c) as a DM client performed in service control SLEE. DM For API412, in order to supply a common message set to all DM clients for the interface connection with data control, the DM client 410 is DM. The demand to data is performed using API412. Moreover, DM Although API412 encapsulates the specific place or which data are needed from DM client, this is because this data may be stored only during the local cache 415 or the DBOR extraction 427. The DM client 410 requires data by the logical name, and is DM. API412 determines whether it is necessary whether the data can search from a local cache, or to require data through DM server from DBOR extraction. every by which the local cache 415 is supplied into the control server 405 -- it has one or more local

aches to application which is [ cache / routing manager / that it is an usable share cache, i.e., for example, a 1-800 process cache, ] different to each process which operates on SLEE, and in which each supply cache has each cache manager of itself is desirable

0107] When the DM client 410 requires data, it is DM first. It checks whether API investigates the local cache 415 and the demanded data are stored there. It is DM when the demanded data are stored in the local cache 415. API searches the demanded data, and a hashing key and an algorithm, or an indexed sequential access method uses standard data retrieval technology, and it supplies it for it to the DM client 410.

0108] When the demanded data are not stored in the local cache 415, the related cache manager 420 searches data through the DM server 425 from the DBOR extraction 427. Especially, it is DM. Notifying that API412 needs specific data for the cache manager 420, a cache manager answers by transmitting a demand to the DM server 425. Then, the DM server 425 uses the DBOR extraction manager 426 for database access, and searches the demanded data from DBOR extraction. The DM server 425 returns the cache manager 420 the demanded data, and a cache manager is DM about data. The DM client 610 is supplied through API412. Moreover, although a cache manager writes the demanded data in the local cache 415, this is dependent on the local cache strategy depending on both function of a computer in which a service request and they are operating, i.e., memory space. These use is gained from the service profile and computer profile which are generated by service administration. As for the data cache manager component for DM400 of IDNA/NGIN, it is desirable to use "client side cache" strategy by each service node.

0109] The IDNA/NGIN Network Operating System (NOS) component 700 is explained more to a detail with reference to drawing 8 -10. As mentioned above, an NOS function is equipped with the InterProcess communication function to the ISNA/NGIN system 170, an object KONEKUTIBITI function, and a local and the whole network resource function manager. Since all NGIN processes perform on various hardware in the architecture distributed broadly, and an operating system platform, NOS realizes platform standalone version communication and place standalone version communication between [ all ] processes. Especially NOS is equipped with some functional sub components, and offers the interface containing service control, service administration, and the interface between data control between [ all ] NGIN processes. Moreover, NOS is an interface between switch call control and service control (drawing 5 ), and is carried out as [ communicate / mutually / two or more processes of operating on the same SLEE ].

0110] As shown in drawing 8 -10, an NOS functional sub component 1) the logical name to a data object and a service object The computer by which the demanded object is operating (as a network address) Both memory addresses The same conversion decomposed into the physical address to discriminate (NT) A process 570 and; 2) The status of all the service node resources in the local resource management ("LRM") processes 575 and 577 of pursuing and maintaining the resource status currently performed by SLEE and the service node, and the whole; 3) NGIN network is maintained. by the global-network resource status ("NRS") process 590 and the; 4) common object demand broker architecture (CORBA) correspondence ORB which offer InterProcess communication A computer platform which offers object KONEKUTIBITI, such as what is offered, and is different, 1 set of services which enable communication between the objects over an API message set and Internet Protocol (IP) communication by the above method of satisfying a specific real-time call-processing operation top affair are included. For example, the typical response time which processes a typical 1 -800 number "collect call" event must be about 50 to 100 msec.

0111] The NOS component 700 can be implemented using the CORBA correspondence ORB to object KONEKUTIBITI which Orbix (registered trademark) offered and IONA of MA, Cambridge and Ireland, and Dublin developed so that it may explain here. ORB offers communication between the objects over a different computer platform by offering the name service which enables matching of the logical name to a physical address.

0112] At the time of system boot, SLEE450 is started and starts an instance with the NOS client component 558 and the service manager process component 554 in the environment. SM SLP554 searches the logical name to other components from the configuration file 580 containing the logical name of the service by which instantiation is carried out immediately. Then, the logical name is supplied to ORB name service, and ORB name service matches the logical name with a physical address. ORB holds service object KONEKUTIBITI from this time. Moreover, ORB service is used also to registration of other services. Each service started on SLEE registers itself into NOS, and ORB discriminates the physical address to a logical name by those registration.

0113] Since the platform standalone version communication between interactive-mode objects is implemented, although an interface is defined, this is possible by the interface definition language ("IDL"). Although CORBA is supporting IDL now, as long as the operation top affair over real-time call processing is satisfied, other object-oriented communication technology, such as a remote method call (RMI) protocol, is able to be implemented. Especially, the interface to each of a NGIN component is defined at the time of the creation, and it becomes usable at the time of operation by storing them all over the continued-indefinitely type datastore related with the local LRM575 which is illustrated by drawing 9 , or a library (un-illustrating). Service is made possible [ performing an inquiry to this library ],

nd learns about a new object interface. The NOS client process 558 and the NOS master 560 ( drawing 8 ) are NOS which is used for the interface connection with NOS service, and is explained in full detail below. In order to call NT and LRM service, it is the NOS class library used by all services that operate within the SLEE concerned.

0114] If drawing 9 is referred to here, the functional architecture of the NOS functional sub component 570 and the LRM functional sub component 576 which reside permanently on the computer which performs one or more SLEE(s) 50 and 450' is illustrated with NT and the LRM sub component which were related with each SLEE. drawing 9 -- specially -- each -- every -- SLEE -- a component -- 450 -- and -- 450 -- every -- NT -- a function -- a factice -- -- a component -- 570 -- and -- 570 -- ' -- and -- every -- LRM -- a function -- a factice -- -- a component -- 575 -- and -- 75 -- ' -- containing -- each -- NOS -- a component -- 700 -- and -- 700 -- ' -- implementing -- at least -- two -- a \*\* -- computing system -- 440 -- and -- 440 Although single SLEE performed on an individual computer is illustrated, two or more SLEE(s) can operate on the same computer in a single site. every -- operating on SLEE450 and 450' -- S1 and ... it is two or more objects or processes which are made into S4, and there is a case of SLP, LLP, CLP, ELP, FD logic program that is operating permanently and the NOS client object 558, or other processes

0115] It is each NOS so that it may explain here. The NT functional sub component 570 and 570' are because a common logical name with single not continuing not changing in whole bar SHON and the whole instance from which process is enabled to call on other processes especially, and the called process differs including the process which discriminates right bar SHON of the data object used or a service object is used. Therefore, NOS The NT component 70 encapsulates refer to [ of the instance from a process ] the object, bar JONINGU, and a physical place.

0116] Each local resource manager ("LRM") component 575 of NOS700 in each service node and 575' determine which service is performed on which SLEE of a node according to the configuration rule contained in the service profile (configuration) file 580 so that it may be explained here. The service profile (configuration) file 580 may also contain the contents of the service profile which is arranged from SA and stored in a local LRM cache. LRM determines on which specific SLEE for this service profile file 580 first stored in the local cache in the node concerned to be read, and to perform service according to the rule in a service profile file, which service to operate actively considering as a permanent object) in SLEE, or whether instantiation is carried out only by accepting a demand.

0117] In a suitable operation gestalt, LRM575 enables the operating-time configuration of service execution, and optimization by pursuing the stability and the state of each service control resource. Especially each LRM functional sub component holds the present loaded condition (processing capacity) of SLEE in the node concerned according to the list of all services programmed so that it might operate on the SLEE concerned, and which service process (refer to object) is operating actively on SLEE and a predetermined threshold.

0118] If it furthermore describes in a detail, the SLEE(server) LRM components 575 will be 1 set of libraries built in the local cache of refer to the object corresponding to each object in a system (logic program), and, as for the refer to the object, an IP address, a port number, etc. will make communication possible including the information about a server. They will be registered into NOS if a new object becomes usable within a system. That is, for those reasons, referring to the object is created, and is registered into a local cache by data control.

0119] NOS after determining instantiation of which service perform an inquiry to the service profile (configuration) file 580, and is carried out immediately The LRM component 575 is NOS about a service active-ized demand. It transmits to the active service manager object 554 in SLEE through the NOS client instance 558 currently similarly performed in SLEE from NT570. The SM object 554 is an API object which enables control of SLEE service. For example, when the demand to inactive service is received, the function which carries out instantiation of the new service is offered. That is, when instantiation is carried out, it is possible to assign a process thread to an object, and service registers itself into NOS through LRM575. If service is called by another service using the logical name, LRM sees the rule in a configuration file, and since the logical name is matched with the physical address of an active instance for which instance is called, it will determine by using ORB name service.

0120] The site LRM577 which operates on individual computer 440"NOS component 700' a top or on share computers, such as a computer 440 or computer 440', is related with a NGIN site or the service node 45 so that it may illustrate to drawing 9 . a site LRM577 -- 1 every -- every which is the function of the present load of all the processes that operate on SLEE -- the usability of service by SLEE -- pursuing --; -- further -- 2 each exception SLEE It functions in maintaining the resource status list with which LRM575 is updated actively and which is a copy with the SLEE identifier to each resource. The site LRM sub component 577 determines which instance of the service demanded based on either of two or more error criteria should be used. Although two or more error criteria are not limited to it, they contain the system and process load of order-of-approximation [ of the data control data needed by the service with the service instance currently called one and the service instance currently called called order-of-approximation site the same in whether SLEE is the same or not?);2, and the called service instance ];, and 3 present.

0121] As an example, when a process (for example, S1 in SLEE1) needs to carry out instantiation of SLP and the S4



and it is necessary to perform a specific process (for example, Vnet service), as for NOS, service (that is, refer to the object) always makes a decision about whether to be usable or not in a local cache (for example, inside of SLEE1) first, so that it may be illustrated by drawing 9. When it does not have "refer to [ as which the local LRM575 was required ] the object", NOS looks for the site level LRM577, and decides on a place as concerned refer to the object corresponding to the demanded service. For example, as shown in drawing 9, if it will not have reached even the use threshold if the object concerned can be discovered by SLEE2, and is discovered and it has the capacity for SLEE performing it namely, it makes the service concerned usable by carrying out instantiation of the instance of the object concerned.

122] furthermore, it illustrates to drawing 10 -- as -- every -- every to SLEE -- LRM577 and LRM577 to each site -- in addition, the NOS component 700 contains further the network resource status ("NRS") sub component 590 which is the process which performs the whole network resource function manager Especially NRS contains the subset of the data held by each site LRM (for example, sites 577a-LRM 577c corresponding to each sites 440a-440c of drawing 10) in a network. NRS590 -- list; 2 of 1 SLEE -- which kind of service -- every -- whether it is programmed so that it may operate on SLEE;; and 3 -- which service -- every -- whether it operating actively on SLEE and the present load of SLEE in the percent base are included This NRS sub component 590 is the logical package-ized function to give the propagation of another level to the demand which cannot satisfy Sites 577a-LRM 577c to NOS. Furthermore, SLEE shows whether facing up, facing down, and the service use threshold reached by the SLEE concerned including a binary indicator [ as opposed to each SLEE450 in the NRS sub component 590 ]. If it is used in order to determine whether the "facing up" or a "downward" indicator, and use threshold application are usable in order that SLEE may receive the service request from other services, and there is application of these indicators and a threshold, a NRS sub component can offer a binary indicator with usable SLEE simply. Although a required SLP object is in SLEE, when there is no capacity which carries out instantiation of the process as which the SLEE was required as an example, the use threshold to the SLEE concerned reaches and a notice is transmitted for the purport which cannot process the further demand to the service concerned to a site LRM577. Moreover, this information is spread in the NRS component 590 ( drawing 10 ).

123] If drawing 8 is referred to, a NGIN system will implement surveillance 595 and will supervise memory space, database capacity, the length of the queue to a required object, the amount of time in a queue, and other resources / load parameters to each SLEE in a system. These elements become usable to NOS700 which makes a decision about the use threshold of SLEE or more based on one of the elements of these. In addition to a fixed threshold, two or more thresholds are usable because of a hysteresis.

124] The example on explanation of the resource function manager performed by NOS including NT, LRM, and NRS to which NOS700 enables offer of place standalone version processing and platform standalone version processing, and optimizes the overall processing facility of NGIN is further explained to a detail with reference to drawing 11 (a) - drawing 11 (b). In the LRM process flow 801 explained with reference to drawing 11 (a) and drawing 11 (b), the service S1 performed on SLEE1 on the service control server 1 needs to call service S2, as a process 802 shows. Service S1 is FD which received the event service control demand from switch structure call control, or a service logic program, for example, in order to complete call processing, it needs to call another SLP and S2.

125] When drawing 11 (a) is referred to especially, service S1 is SLP. A demand is published to NOS700 using the logical name to S2. If the SLP demand to a service object is received, NOS name conversion function 570a will be performed as shown in a process 804, and it will be determined whether NOS has "refer to [ which recognized that required service operated actively on the local service control server 1, namely, was related with the logical name of required service in it ] the object." As for the data stored in a local server cache, it is desirable that the following NOS naming data fields are included. The SLP logic service name which is the name which is a logical name explaining service and functional discriminator data point out typically; 1) As opposed to a specific customer or a specific node; 2. which requires the version concerned to which 2, for example, service, operates Version number; 3 arbitrary arrangement explaining the version of the specific service which is in the case of being needed ( ) Although SA has arranged the work package to the node, actively service Namely, when having not turned, Active (that is, it is shown that service is active now) or a fall back (for example, in order to realize high-speed reversal) The IP address which discriminates the physical place of a status; 4 object instance including the case where to return to the version in front of a service object is wished, A port and the object name which can include the information on other or reference; 5 operation time, and pause time; 6, for example, an object, [ which is not usable ] Or the fall-back object name performed when it is in error process object name [ when active-izing is improper ];, and 7 fall-back status. A local server NOS naming process uses the service offered by the LRM status processor (un-illustrating). A LRM status processor updates a local server cache status database only about the service active now which is operating in SLEE of inpointing in a service control server. Therefore, a local server NOS name conversion function can be performed

ically first. If NOS obtains a name demand first, in search of a logical name, an object name (or refer to object) will be acquired. NOS acquires an object name from a logical name, and as a process 806 shows a node LRM process, the best instance of the required object processed based on one or more business rules is determined.

0126] In a process 804, a logical name is recognized, when referring to the object is usable, processing progresses to the LRM function of a process 806, and a use threshold etc. determines the active ("usable") instance of S2 which operates on SLEE1 in accordance with a specific error criterion. When an active instance is not discovered at all, LRM can be investigated in order that S2 may check whether it is programmed on SLEE1, nevertheless whether instantiation is carried out. If there is usable capacity with sufficient SLEE1 when that is right, NOS700 can determine to carry out instantiation of the instance of S2 on SLEE. As mentioned above, it is that LRM of server level is only conscious of that by which instantiation was carried out by the server being conscious of the active thing. An object is active now, and when instantiation is carried out on local server level, referring to [ which carries out instantiation of the new thread for this service ] the object is returned to an SLP demand. NOS returns "refer to the object", when service which started the instantiation of a new service thread and was demanded based on refer to [ which was returned ] the object is performed and instantiation is not carried out yet.

0127] In a process 804, when SLEE1 does not have sufficient usable capacity, or when not usable because of operation on SLEE1 of S2, in a process 810, LRM on SLEE1 transmits a service request to site LRM577a ( drawing 0 ). Site LRM applies the same business rule and determines whether the instance of S2 is active on another SLEE of the site concerned, or instantiation should be carried out. Therefore, in a process 810, a node NOS name conversion function is implemented, and the demanded logical name is usable at the node concerned, or it determines whether old "refer to [ which was related with the logical name of the service as which same or another SLEE at a different local service control server was required by this \*\*\*\*\* this node ] the object." When the logic service name is recognized in a process 810, the NT sub component 570 is NOS. An inquiry is performed to LRM575 and it determines whether to use the instance of S2 throat. Then, in a process 814, Node LRM is returned to SLP which applies a business rule to a node cache status database (un-illustrating), searches "refer to [ of the request to the service demanded when active ] the object", and is calling the address (drawing 11 (a) process 802). When it is determined that instantiation of the service with which the present instantiation of the service is not carried out or which was demanded in specific SLEE is not originated and carried out to a processing load or other restrictions which are imposed, assignment and loading processing are performed in a process 818. This is performed by investigating a node cache status database, for example, implementing the applicable business rule about service order of approximation, data order of approximation, a threshold, the present processing load, etc., when it is determined that instantiation is possible for a service object, it carries out instantiation of the service demanded in SLEE, and it returns it to SLP which is calling the address. In addition, in order to determine instantiation of which service thread is carried out when many services are more nearly usable than one to the instantiation for every SLEE, round robin scheduling is an execute permission.

0128] If it returns and refers to to drawing 11 (a), when the logical name as which the present node was required will not be recognized in a process 810, Namely, do not have "refer to [ which was related with the logical name of the service as which the node cache was required ] the object", or it originates in the applied business rule. When the instantiation of the object cannot be carried out in a node, it sets at a process 822. An inquiry is performed to the global-network resource status (NRS) process 590, the present status of SLEE in the intellectual network 170 whole is investigated, and SLEE which can process the service request to S2 is determined. As a process 820 is shown before this, in order to conduct investigation and to discover "refer to the object", it is determined whether the index number showing the number of times to which the inquiry was performed to network resource management exceeded the predetermined limitation (for example, 3 times). When this threshold is exceeded, a process is ended and it is notified that a service object cannot discover an administrator and that an error situation exists as shown in a process 821. When it is not over the NRS inquiry threshold, as shown in a process 822, the NRS process 590 determines whether to execute a permission about the service as which which service node in a network was required. After determining the node in an intellectual network, following the process 824 of drawing 11 (b), a process is implemented, in order that node NOS name conversion function 570b may gain "refer to [ which was related with the logical name of the demanded service ] the object." In a process 824, when the logic service name is not recognized by the node concerned, in a process 829, it investigates whether the increment of the NRS querying index number is carried out, a process returns and progresses to the process 820 of drawing 11 (a), an index number threshold exceeds, and an error situation exists. In the drawing 11 (a) process 820, when the NRS inquiry index is not over a predetermined threshold, in a process 822, an inquiry is again performed to the NRS process 590, and the new place of the usable service in another service node is discovered.

0129] If a logic name is recognized in Step 824, processing will continue to Step 826 there and the address relevant to

he object reference demanded according to the processing load which can be received will be determined. This address is returned to SLP demanded as there showed to Step 802 of drawing 11 (a). In Step 826, when instance generation (active) of the service is not carried out till then, processing enables allocation and load processing by progressing to Step 828, checking the node cache state database 768 in the node, performing a business rule, and carrying out instance generation of the demand service in SLEE of the part judged as the ability of a service object to use for instance generation. Then, the address of the object SLP by which instance generation was carried out is returned to a demand client in Step 824.

0130] Selection of the active instance S2 returns the object reference for the S2 instance generation to NT on SLEE1 (Step 802). And NT is efficiently changed into the object identifier for instances S2 which had the logic name S2 chosen, and uses the object identifier for S2 for the mutual processing communication between S1 and S2. An object identifier includes an IP address, a port, and other information that the physical position of an object instance is pinpointed. If an object reference is determined, NOS will bring about the object connectivity in two service compartments there by making the CORBA conformity ORB and data communication connection except a protocol like UDP/IP. If it will be under operation on the same SLEE, or if it will be under operation on thousands of mile SLEE of everything but another site over there, the service position which received the call is completely clear for a call service. If instance generation of the SLP which needs a call is carried out on SLEE of a remote site in this way, this call will still be held on the switch which received this. Preferably, when an object reference is accessed also at once, to other sites which go via NRS level, it is good to guarantee that the cache of NBS is carried out and it is heard to a demand site for the references of the future [ reference / object ] through a service manager. In this way, in the present example, when this service is needed again, in order to reduce a series of table search by starting table search of site LRM, a place does not depend the object reference for service S2 how, but a cache is carried out after that into the local cache of LRM575 of SLEE1. It is clear to this contractor that methods by which a service object reference is applied in SLEE are various. for example, all object references [ in / a site LRM577 / in an NOS data copy mechanism ] -- the site -- setting -- every -- individual to SLEE -- it is prepared so that it may copy to LRM

0131] The context called No. 800 [ one to ] ("18C") explains 18C call processing and a service use scenario to the instantiation purpose below with reference to the flow chart of drawing 13 (a) or 13 (c), and the outline functional diagram of drawing 18. As shown in Step 920, a call reaches the NGS switch mechanism 75 first. If NGS receives a call, a bearer control element ( drawing 5 ) will supply a call control element to the access line which received the call like ANI, a number to be dialed, or other data required for call processing. A call control element maintains the statement model for a call in connection with performing according to the programmed logic. The trigger which transmits a service request to FD510 which carries out instance generation of ELP540 further at a statement model, and as shown in drawing 18 is contained. In order to carry out instance generation of the ELP, the NGS call control element 70 carries out addressing of the message to NOS using the logic name for ELP, as shown in Step 923 of drawing 13 (a). As answer this, a message is transmitted to a service management program object ( drawing 8 ), and instance generation of the ELP is carried out within SLEE and expressed with Step 926, NOS calls the object reference turned ELP, and returns it to a controlling mechanism. A NGS call control element contains an object reference in the require service message transmitted to FD in SLEE as shown in Step 929. In this way, all the proper event data generated by any processings to the call are written in the ELP process by which instance generation was carried out. Especially, addressing of the require service message is carried out to the logic name for FD, and it is changed into the physical address for FD logic program under operation by the NOS-NT element in the same service node as having received the call. The number and ANI which were dialed, and other data are contained in a require service message.

0132] Next, as shown in Step 931, as for FD, which SLP discriminates whether it is going to process the receiving service request using a functional discernment table. for example, when a receiving message is 18C service request, you should process by 18 C-SLP -- \*\* -- it becomes The following table 3 is an example of a brief sketch FD table which has a header containing the pointer to various toll-free call services of No. 800 [ one to ] .

entry port table "001001" SLP pointer "V network"  
 001001" Table pointer to a FGD table FGD table 1800\* table pointer 800 Table 1888\* table pointer 800 Table 1900\*  
 table pointer 900 Table 1\*SLP pointer "a local office number"  
 00 tables 1800 correction SLP pointer to "1-800-C" 18008888000SLP pointer "800 Service"  
 800\* SLP pointer "800 Service"  
 888\* SLP pointer "800 Service"

here, FGD is a functional group identification descriptor.

0133] Especially, based on the part and the received call classification (for example, No. 800 [ one to ]) which the call generated, FD determines a suitable SLP logic name within a line network (switchboard). For example, an identification number "001002" shows the receipt of the call which requires the table search in a FGD table (pointer to



FGD table). The direction of a FGD table maintains the pointer to other tables based on the call number, for example, 00\*. However, "\*" is a delimiter. From these 800 tables, FD receives the pointer to the demanded SLP logic name. Consequently, this SLP is called, a service request is passed to NOS, and NOS carries out instance generation of the object of CLP545, LLPO530, and SLP520 according to 18C service with the demand. For example, about LLPO, the logic name for LLPO is supplied to NOS based on the bearer control line which received the call. Discernment of this circuit is due to either ANI discriminated by the bearer control element 80, or an access line. Also it being the same as the access line in which NGS received the call, although ANI discriminates the original access line which emitted the call, or differing \*\*\*\*. That is, the received call is what was generated for example, on the local network, and may pass the switch 75 on a compatible carrier network. So, it can call and standby, i.e., the circuit related function to be busy, can be discriminated by ANI. NOS changes the logic name for LLPO into instance generation of LLPO to a physical address. Although instance generation of other logic programs (SLP etc.) is carried out in other sites, in a site with the related circuit, instance generation of the LLP can also be carried out by one side. Although instance generation of the SLP is carried out in SLEE, SLEE is on a service control server or a call control server. If instance generation is made, it refers to the data control mechanism for a circuit related function, and the state of a generating circuit is maintained, and all functions, such as call standby and routing by overflow, will be called when these functions are called by call origin (for example, waiting [ call and ]) and the line network (under routing by overflow).

0134] With reference to Step 934 of drawing 13 (a), NOS receives the turnover demand of a service request from the functional identifier containing the logic name showing the specific service called 18C. NOS checks that the demand includes a table in the logic name row in an instance table (not shown), and judges what possible SLP process which meets this service request it has. It checks which instance of the demanded classification NOS should use again, as shown in Step 937 through a NOS-LRM function. And as shown in Step 941, NOS transmits a demand to the service management program object under operation on service control SLEE, and calls demanded 18 C-SLP service. Although NOS chooses from NGS SLP from the service control server which received the original service request announcement which carries out ingress with a desirable operation form, it turns out that NOS can choose SLP in all service control elements based on the lists and those present states of a service control instance through execution of NOS-LRM. As shown in Step 943, NOS directs to SM that an SLP object carries out instance generation, as it judges whether instance generation of whether instance generation of the selected SLP was already carried out and the selected SLP is yet carried out and it is shown in Step 946. Otherwise, when instance generation of the selected SLP has already been carried out, a thread manager assigns a new processing thread to an SLP object, as shown in Step 945.

0135] The next step 949 of drawing 13 (b) requires that the SLP processing by which instance generation was carried out registers the physical address into NOS, and that NOS should assign the SLP to a service request. And NOS passes service request turnover message to new SLP, as shown in Step 951. In parallel to it, NOS transmits all the data containing the object reference for the objects of SLP and ELP by which instance generation was carried out and instance generation was carried out to CLP, and LLPO. CLP and the object reference turned ELP can be supplied to LLPO and SLP again, and, thereby, LLPO and SLP can take now an interface with CLP and ELP. Finally, as shown in step 954, SLP starts call processing here according to the programmed logic.

0136] In the context of 18C call, preferably, 18 C-SLP520 receives required data from 18C routing database (not shown), and makes a suitable decision. As shown in drawing 13 (c), 18 C-SLP520 calls the following steps. Namely, as the logic name which 18C routing data needs for NOS-NT is transmitted in Step 960 and it was shown in Step 962 as it refers for the logic name of 18C routing database to DM, an actual 18C routing DB name and its storing position are received from DM and it was shown in Step 964 Emit a demand to NOS-LRM, judge whether local use of 18C routing database is possible, and if possible As were shown in Step 966, and the physical address of 18C database is returned to 18CSLP(s) and it was shown in Step 968 By transmitting called what [ 800 to ] No., and Circuit ID and the original switch trunk line, and ANI As reference is transmitted to a data control mechanism about table search of a customer profile and it is shown in Step 970 and Step 972 The last routing information containing the switch/trunk line which returns to 18 C-SLP is received from DM by requiring table search of the termination position (node) where the termination specified by the routing response is actual from DM, and receiving an actual termination node position from DM.

---

since it became timeout time, translation result display processing is stopped.

## NOTICES \*

Japan Patent Office is not responsible for any damages caused by the use of this translation.

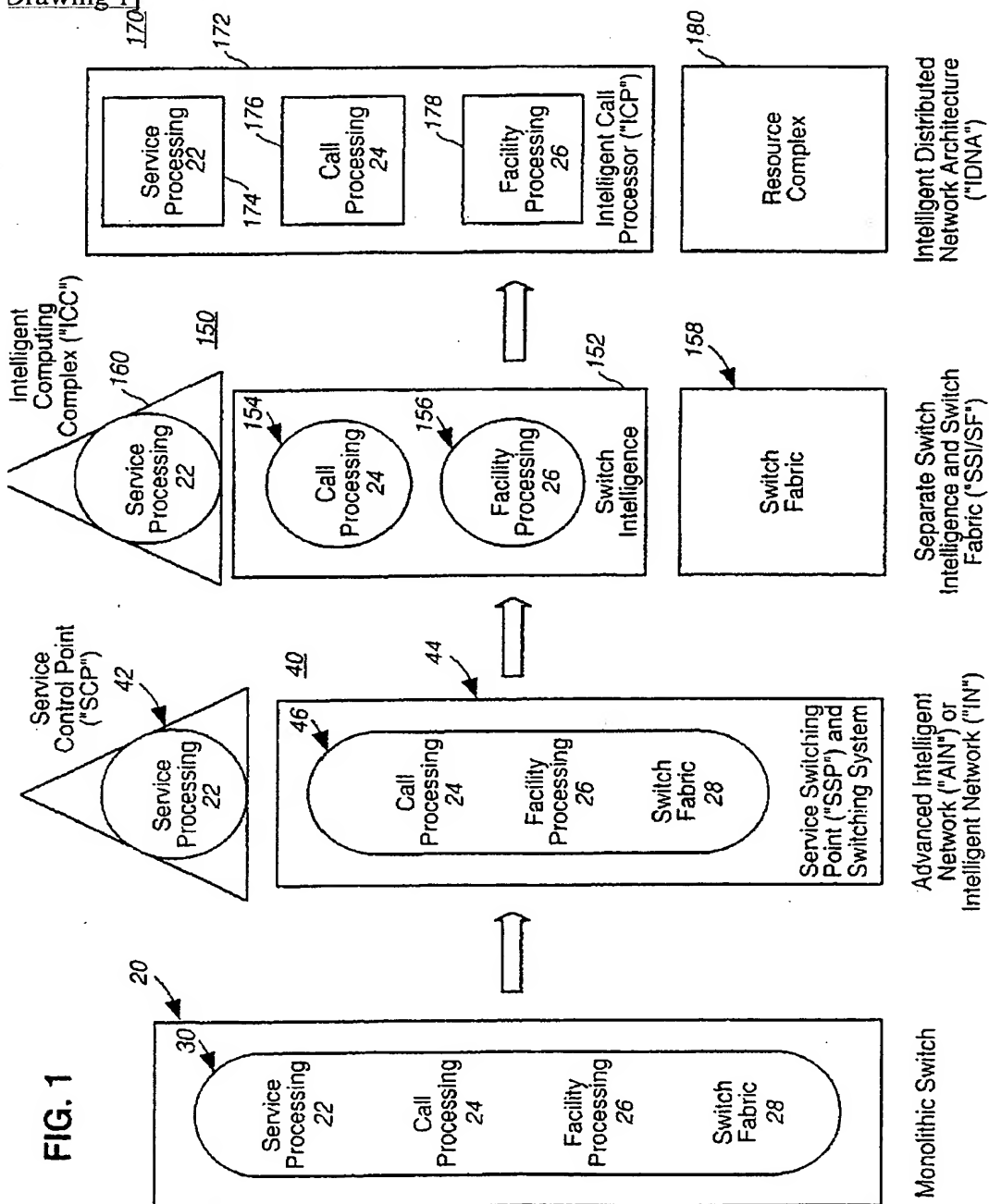
This document has been translated by computer. So the translation may not reflect the original precisely.

\*\*\*\* shows the word which can not be translated.

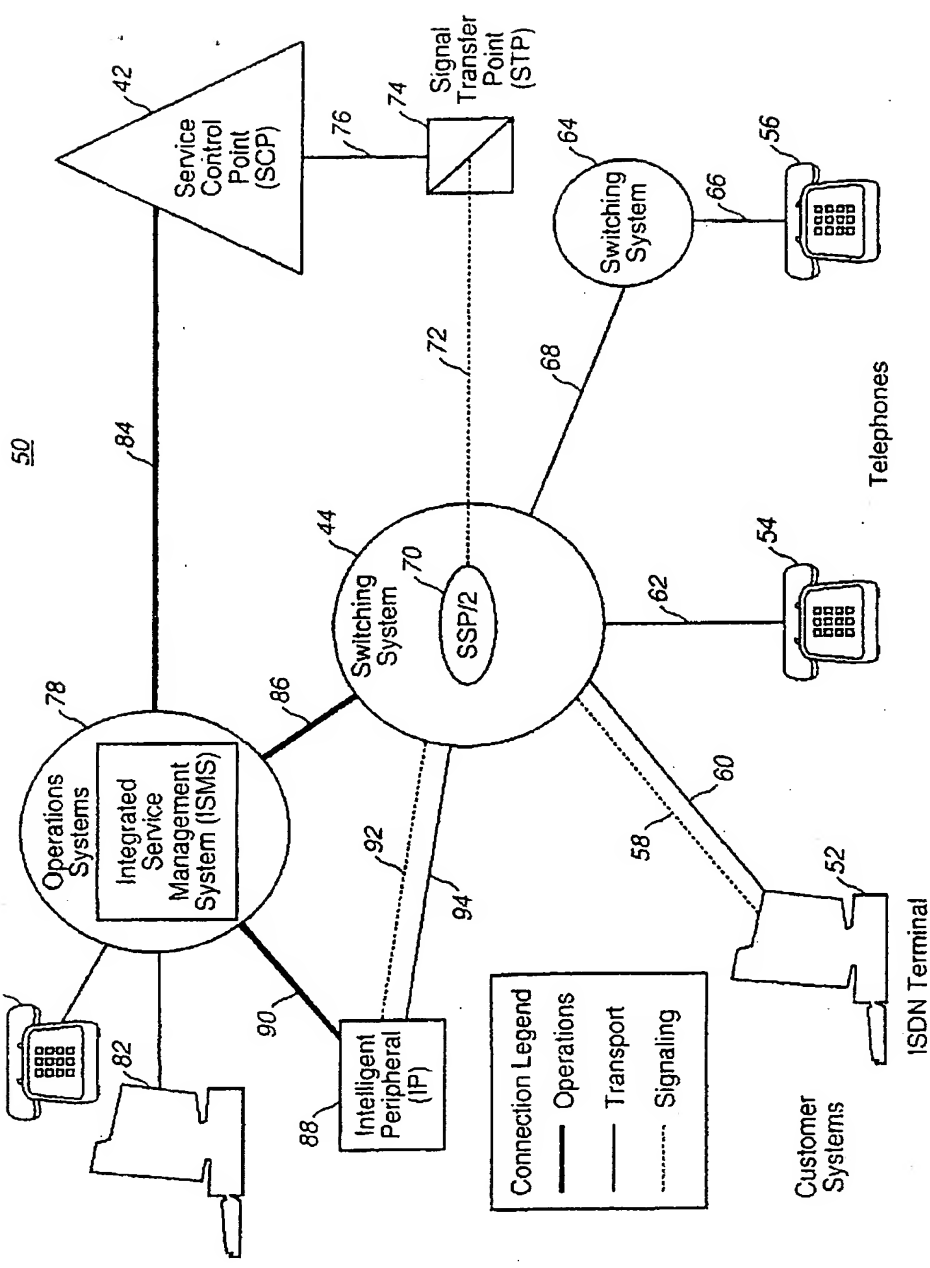
In the drawings, any words are not translated.

## DRAWINGS

Drawing 1]



Drawing 2]



(Prior Art)  
FIG. 2

Drawing 3]

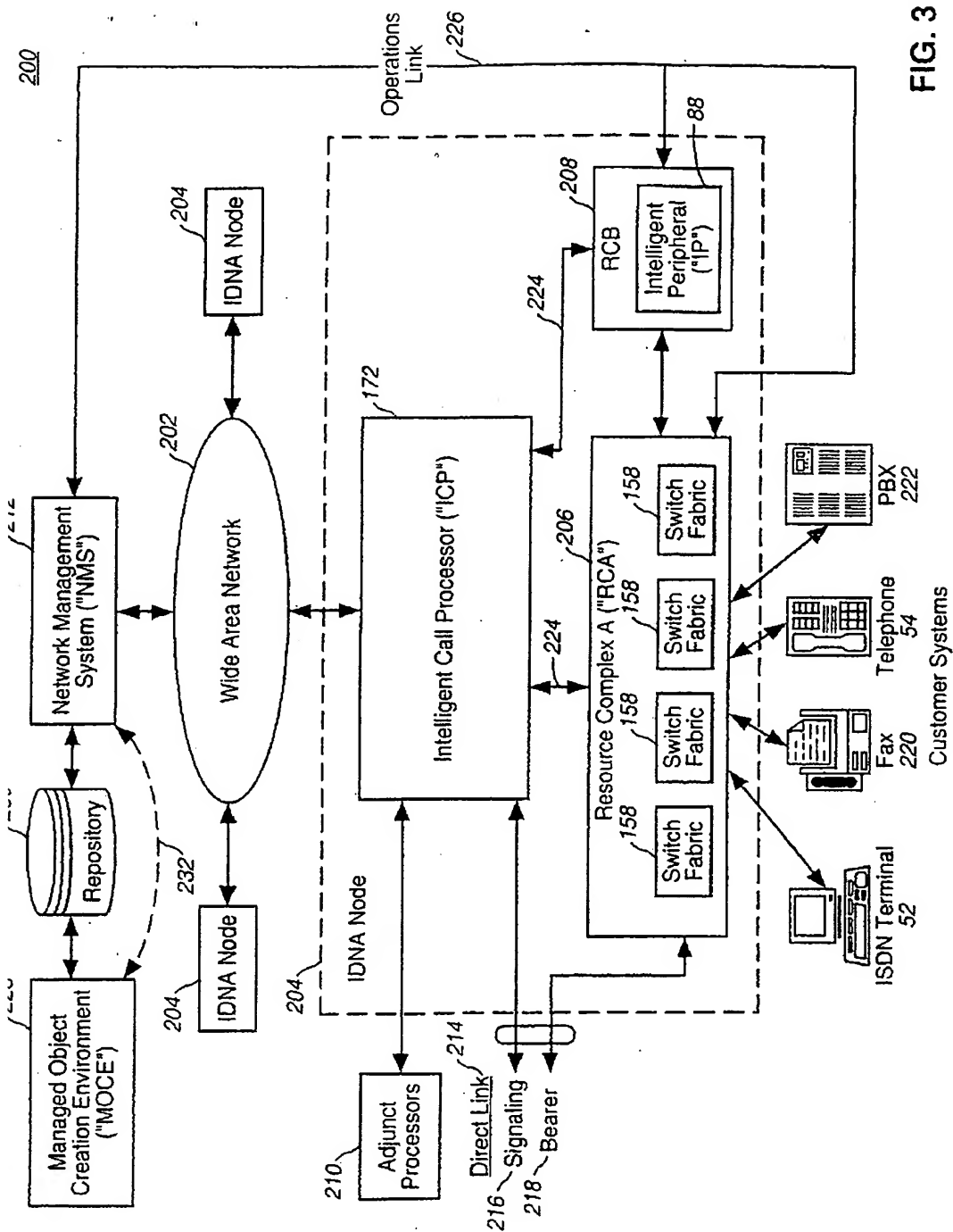


FIG. 3

Drawing 4 a]

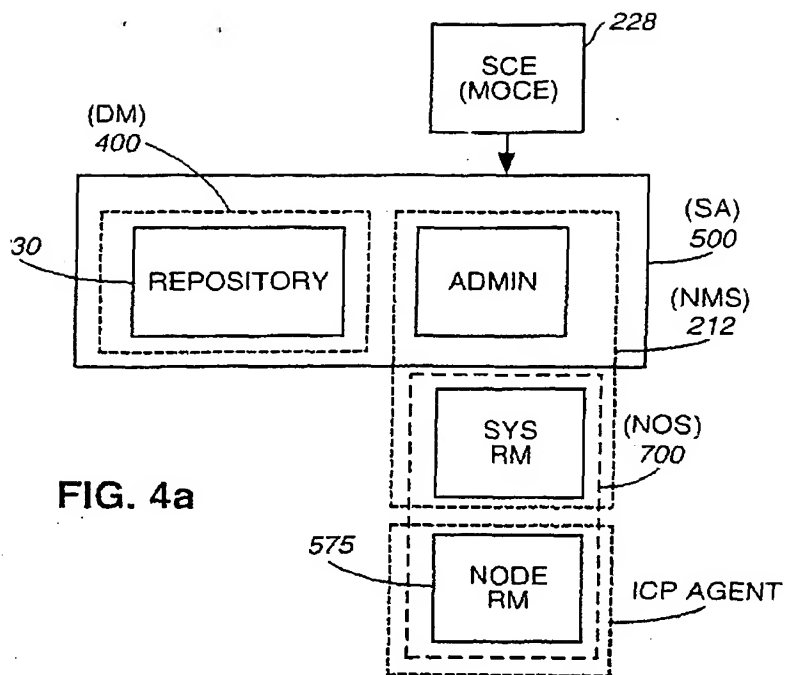


FIG. 4a

Drawing 4 b]

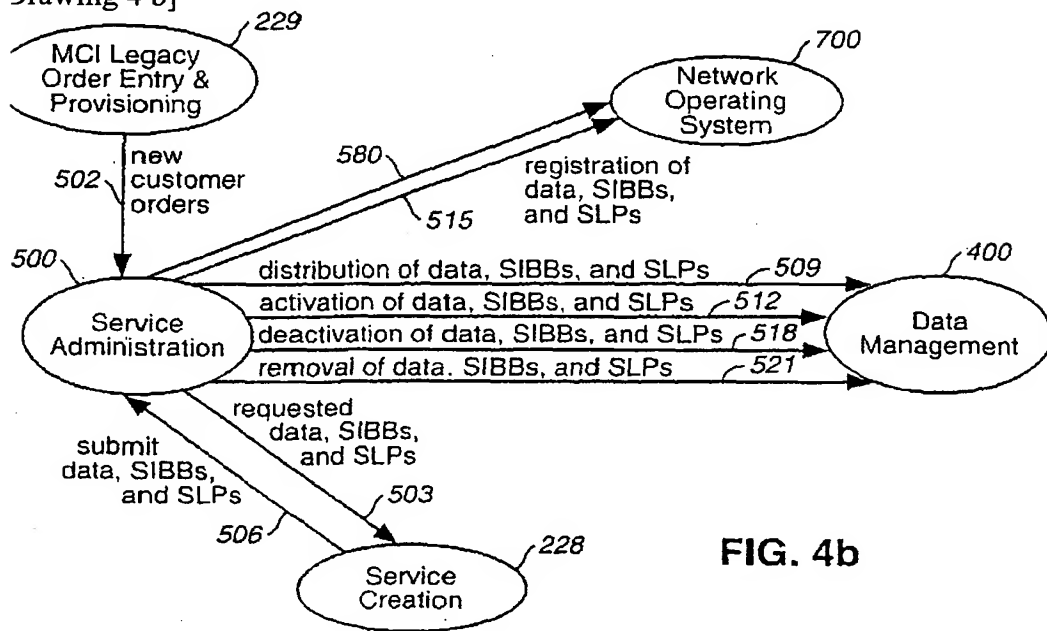


FIG. 4b

Drawing 4 c]

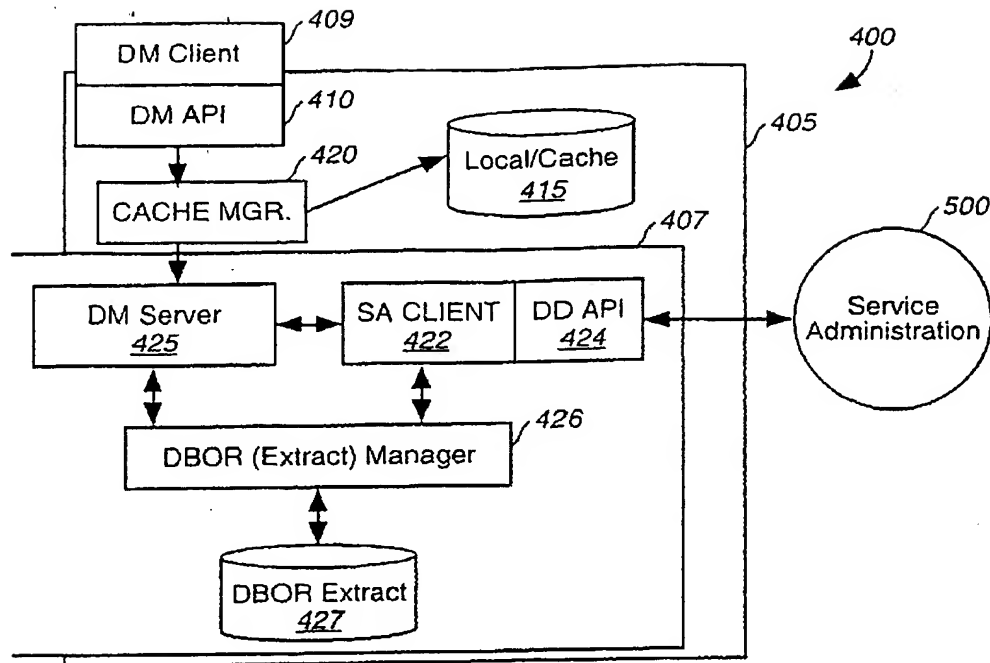
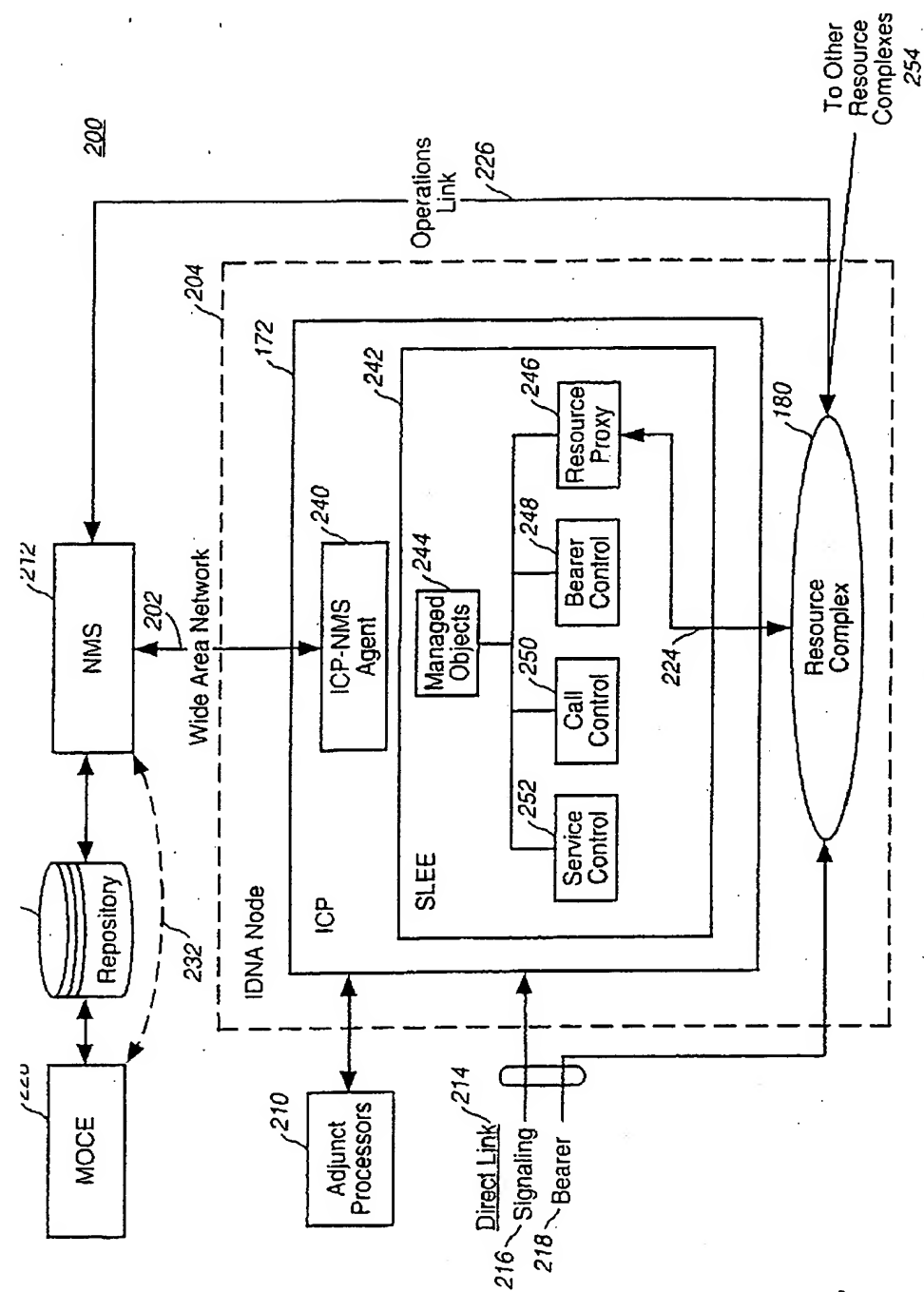


FIG. 4c

Drawing 5]



Drawing 6]

FIG. 6

Drawing 7]

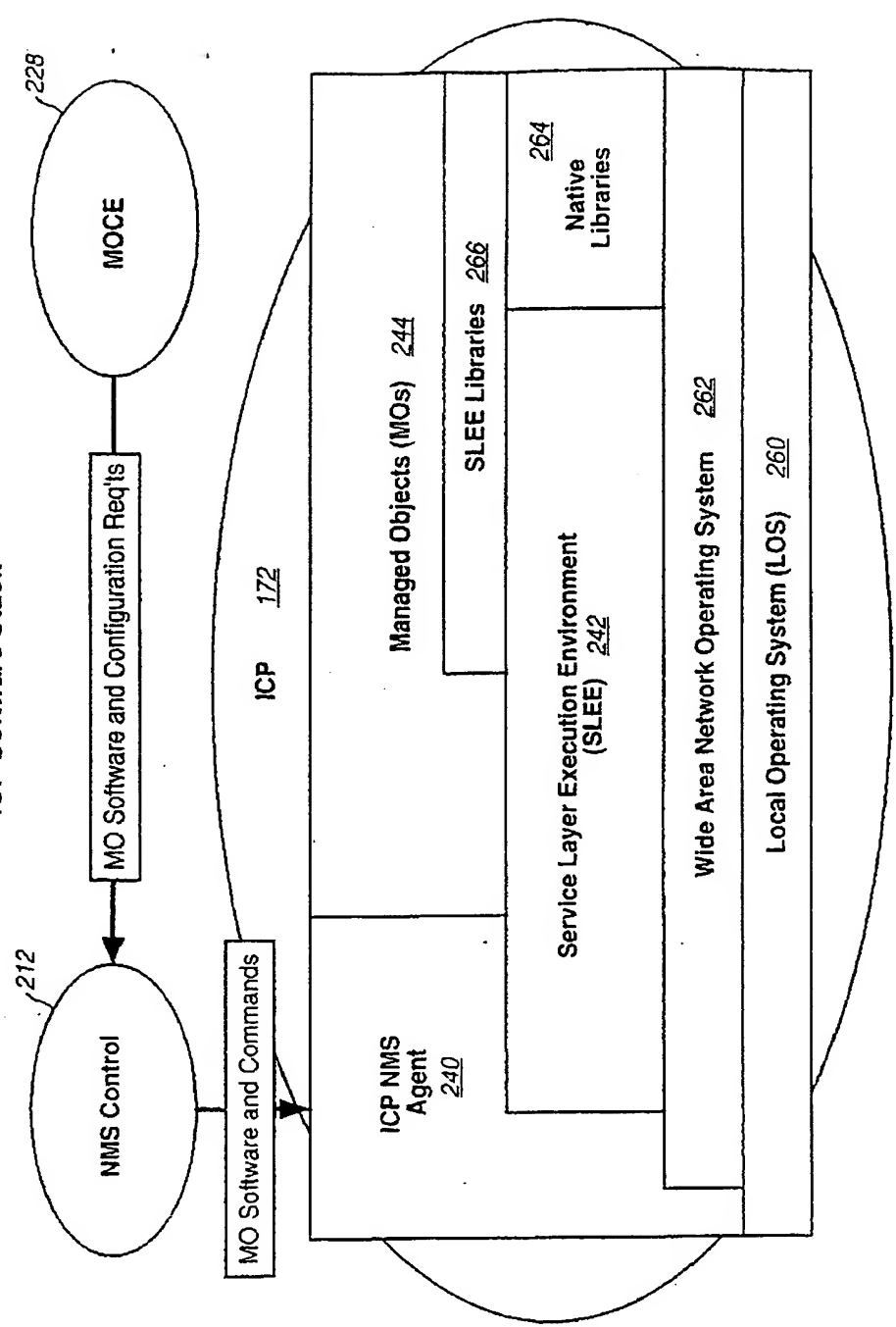


FIG. 6



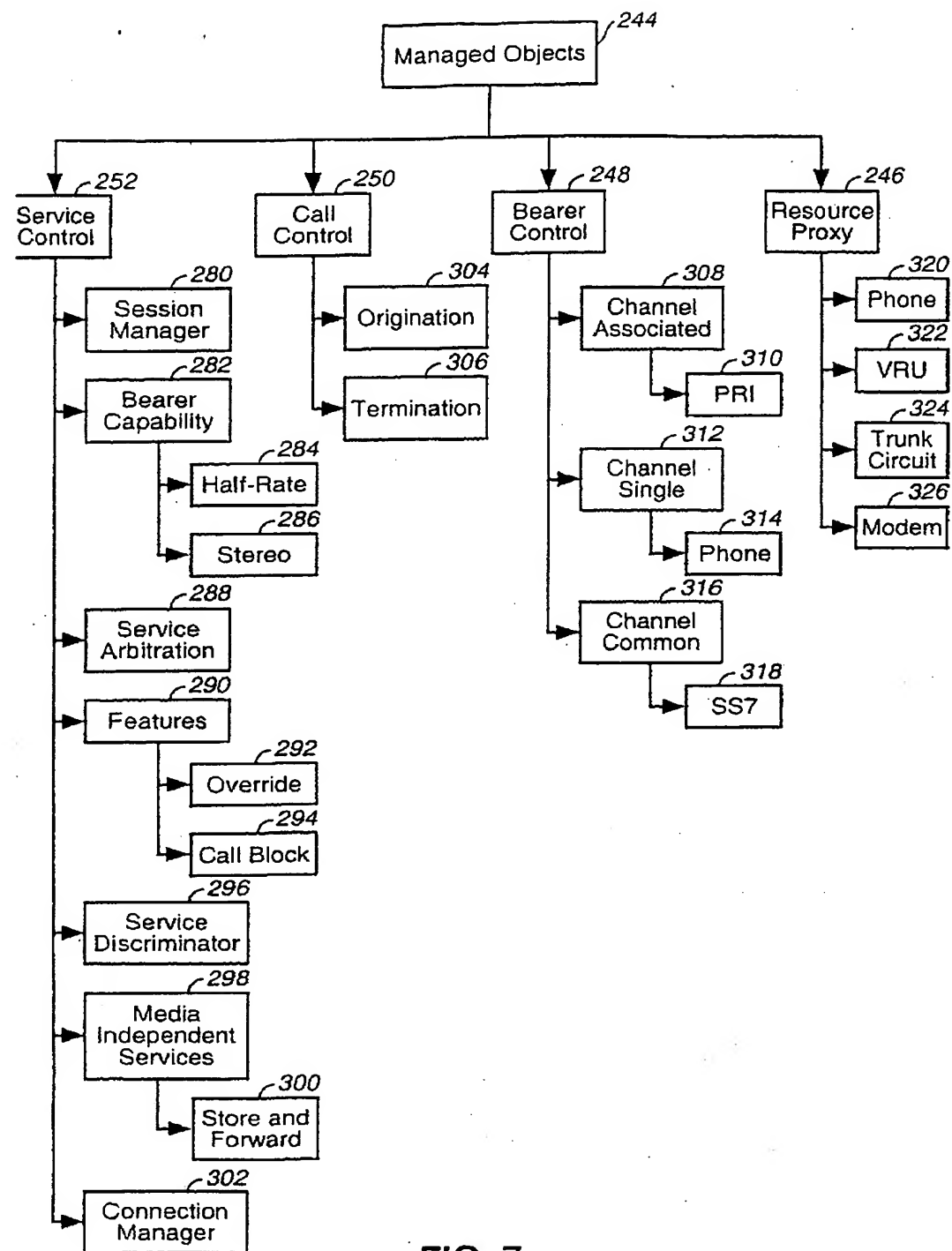


FIG. 7

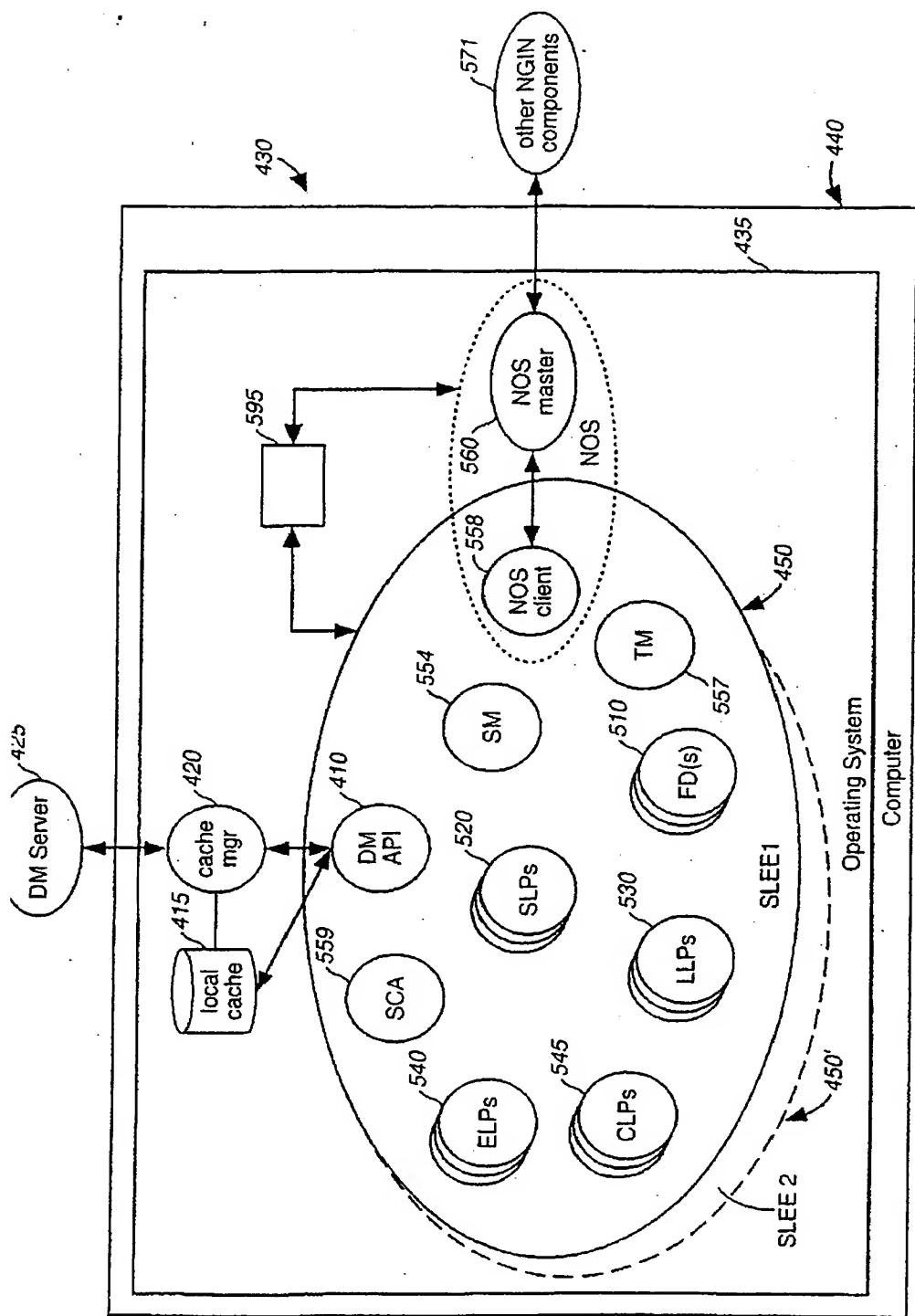


FIG. 8

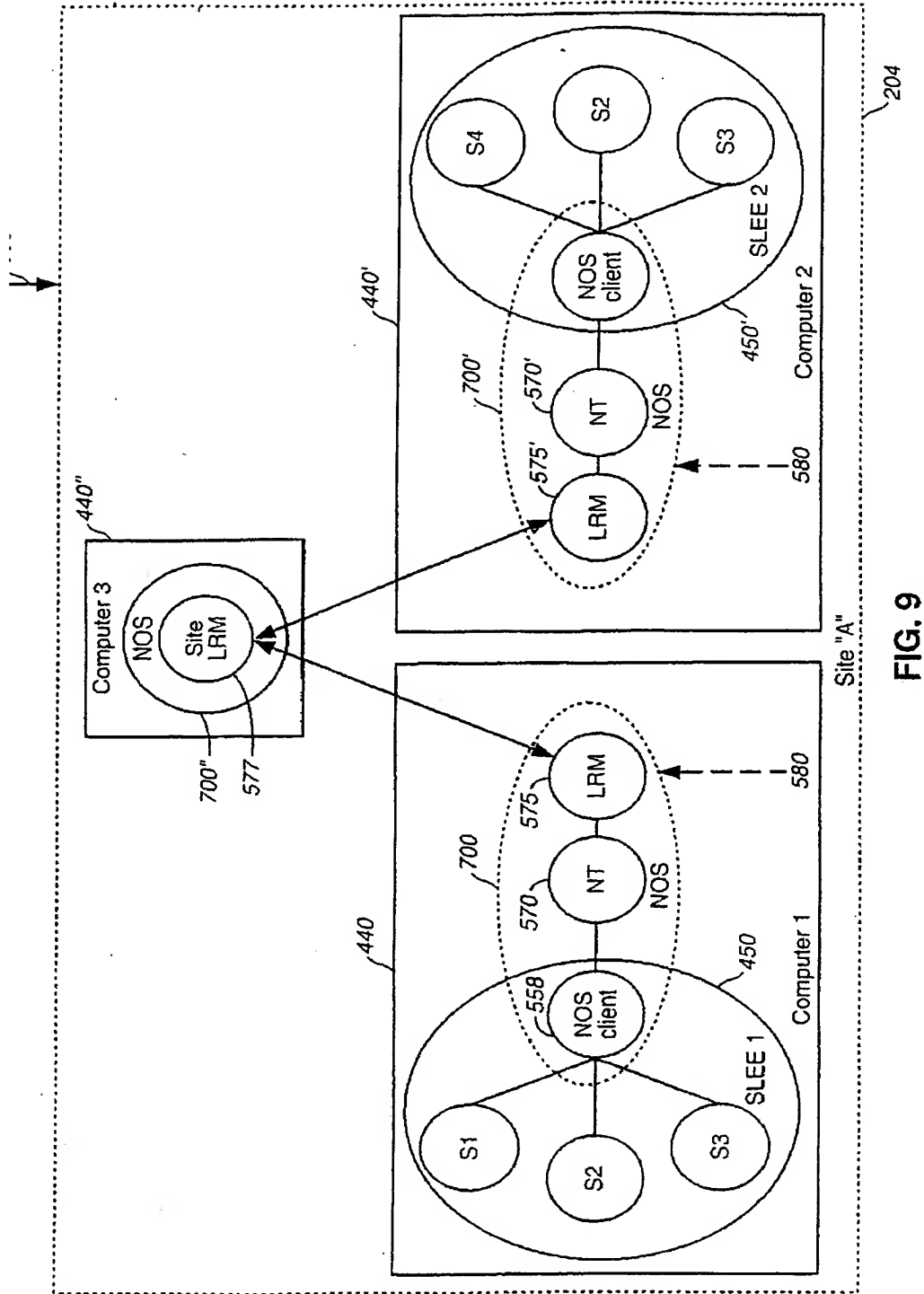
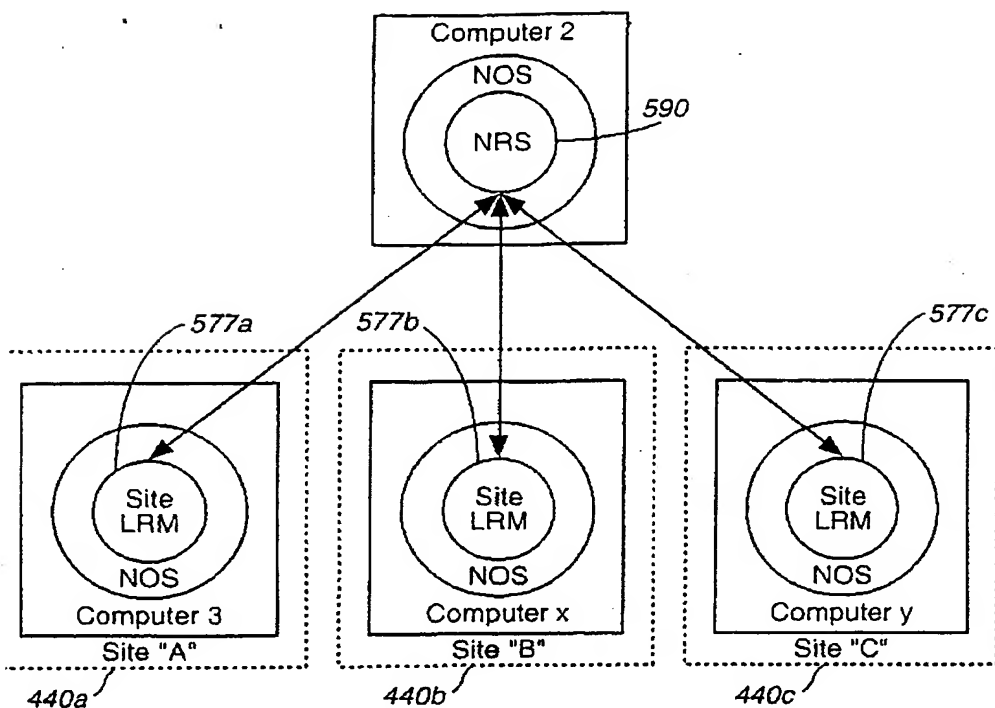


FIG. 9



**FIG. 10**

Drawing 11 a]

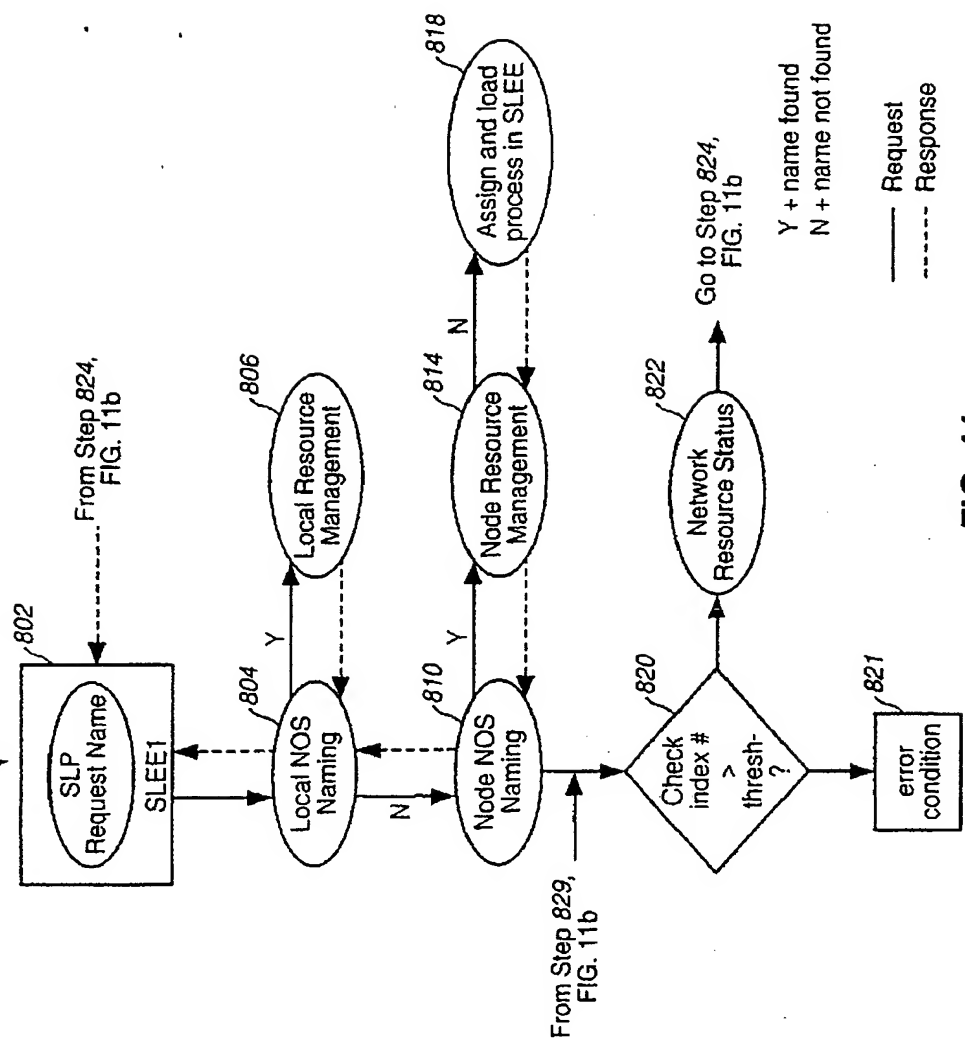


FIG. 11a

Drawing 11 b]

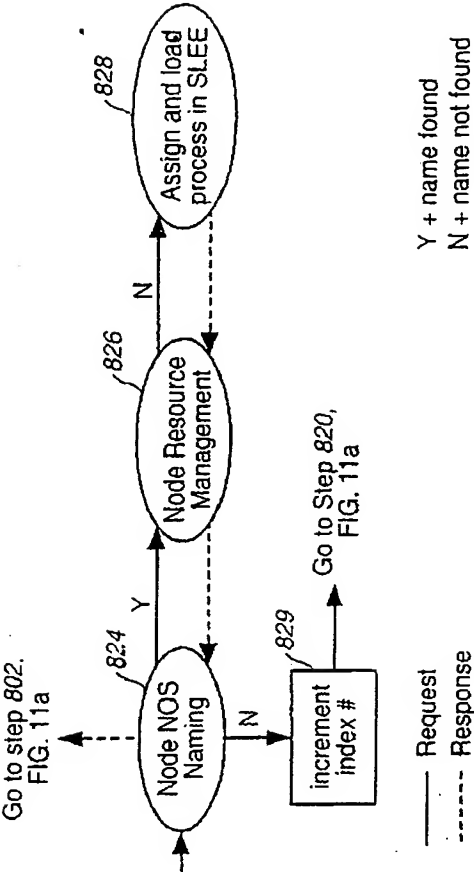
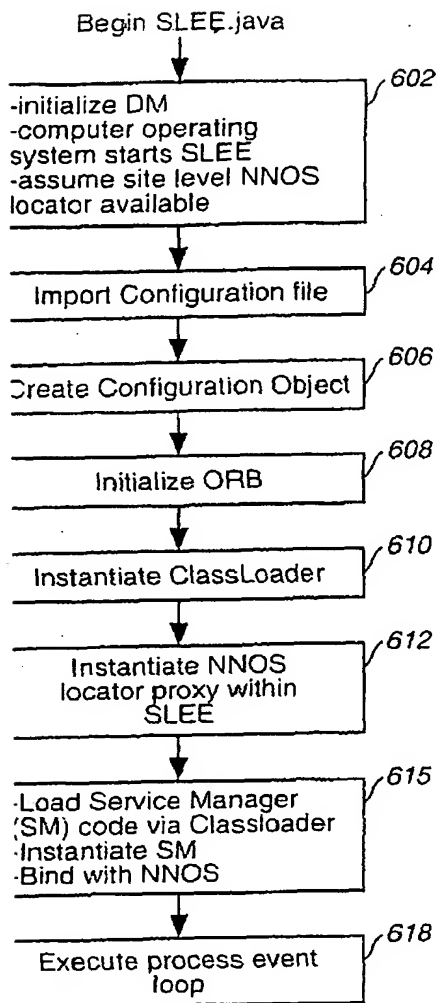


FIG. 11b

Drawing 12 a]

**FIG. 12a**

Drawing 12 b]

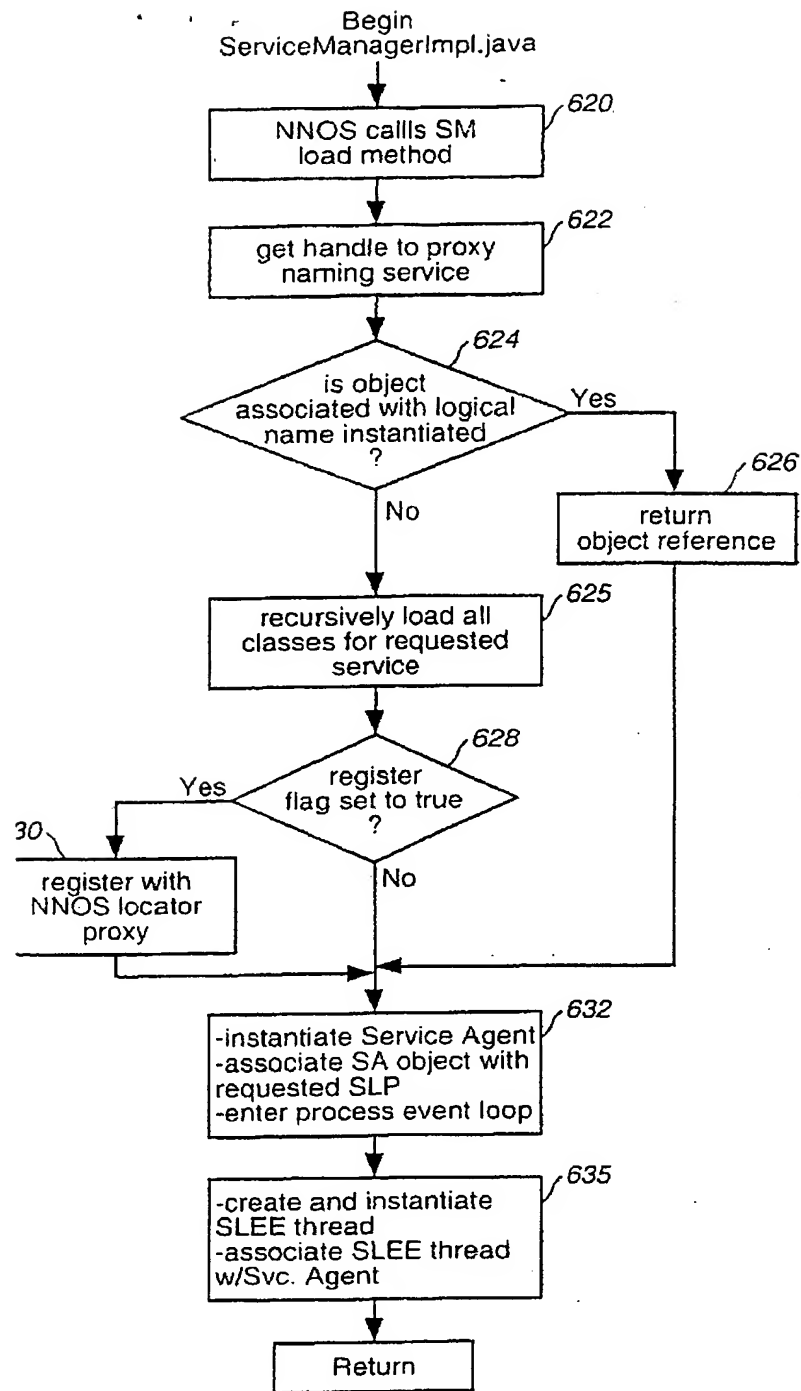
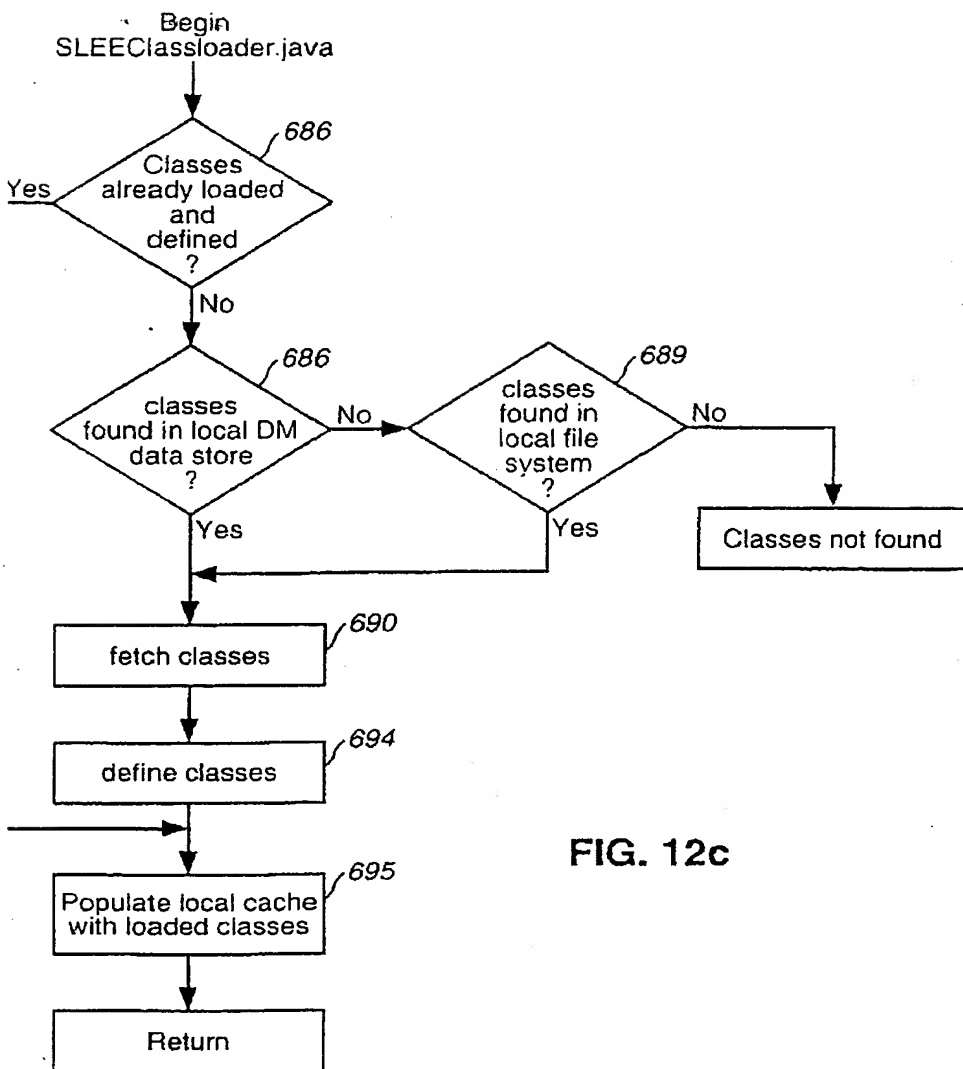


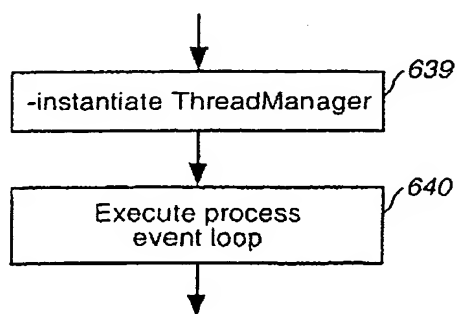
FIG. 12b





Drawing 12 d]

IG. 12d



Drawing 12 e]

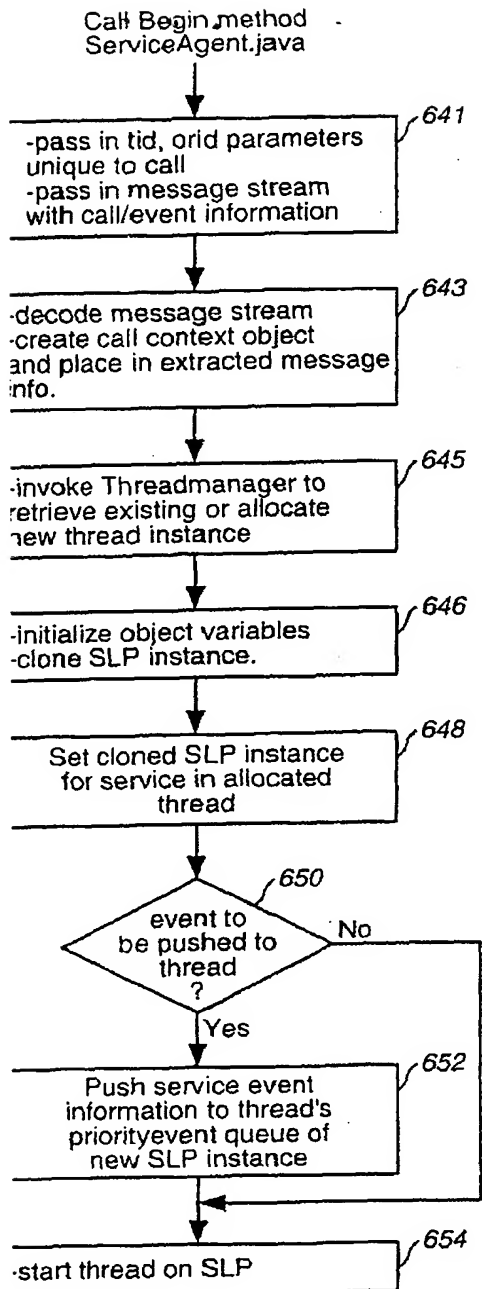


FIG. 12e

Drawing 12 f]

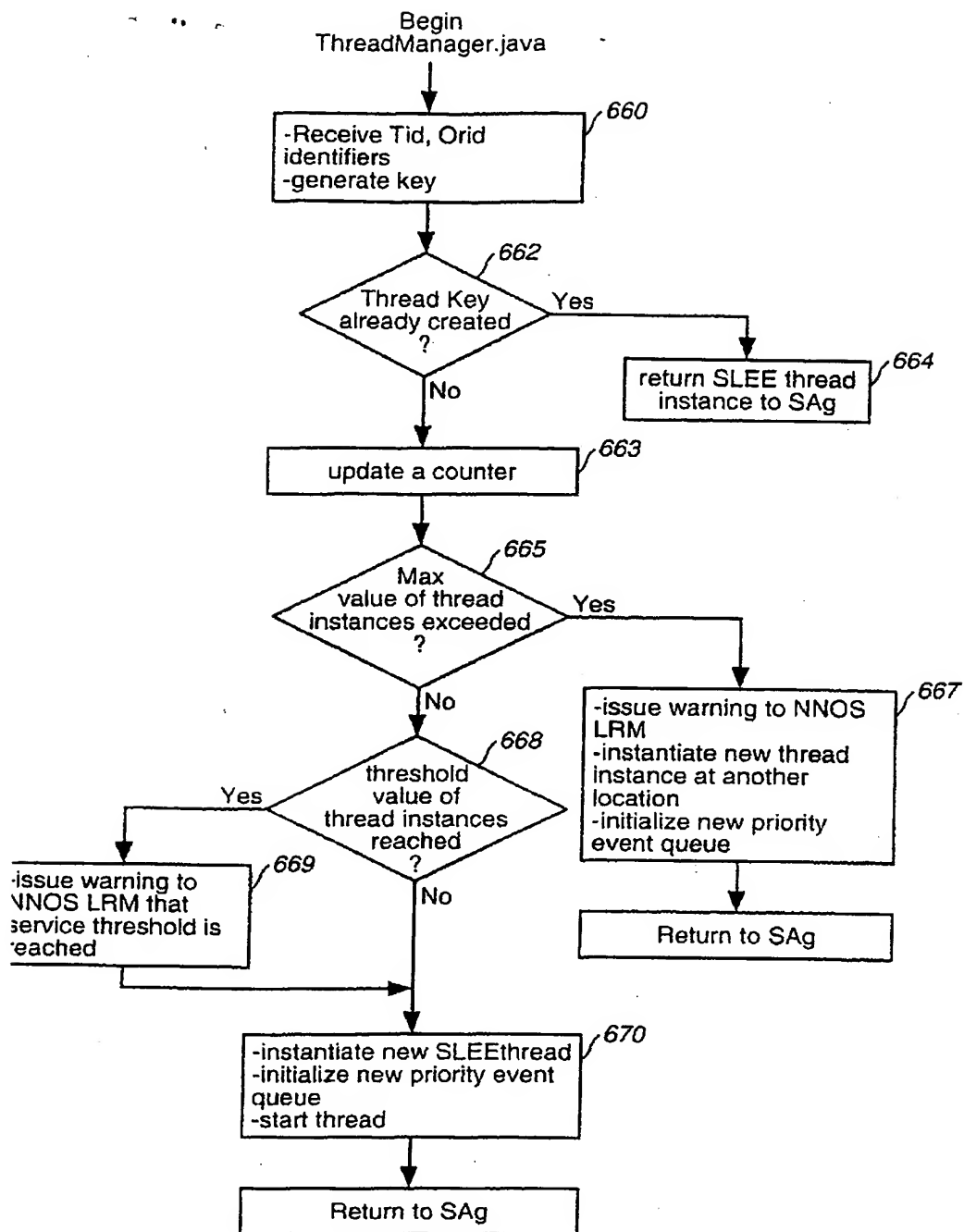
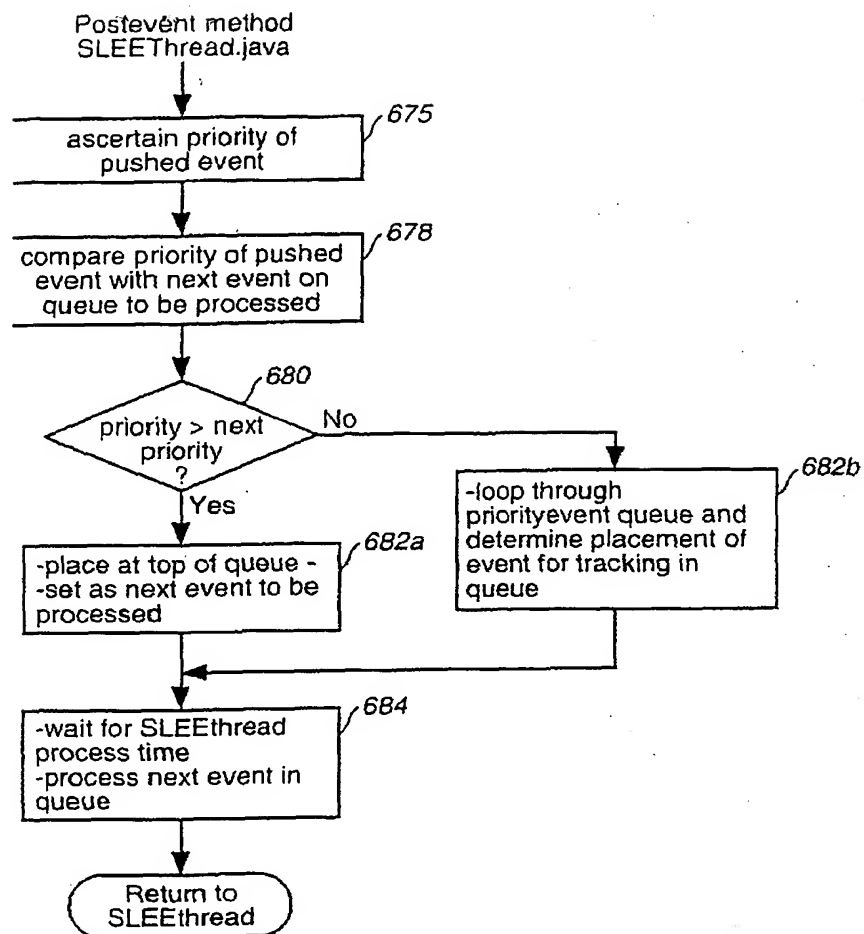


FIG. 12f

Drawing 12 g]



Drawing 13 a]

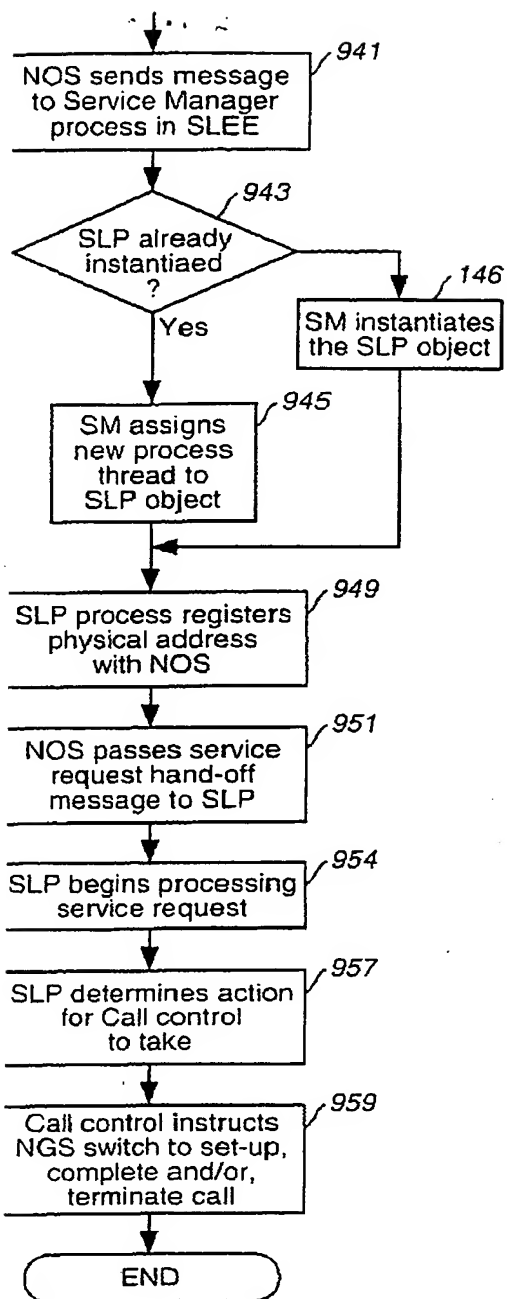


FIG. 13b

Drawing 13 c]